

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN**

CAO GIANG THIÊN TÀI – TÔNG NGUYỄN QUỲNH TRÂM

**HỆ THỐNG QUẢN LÝ MẪU THIẾT KẾ VÀ
THÀNH PHẦN TỪ MẪU THIẾT KẾ
PHỤC VỤ CHO VIỆC XÂY DỰNG HỆ THỐNG THÔNG TIN**

KHÓA LUẬN CỬ NHÂN TIN HỌC

TP. HCM, 2005

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN**

**CAO GIANG THIÊN TÀI – 0112029
TỔNG NGUYỄN QUỲNH TRÂM – 0112328**

**HỆ THỐNG QUẢN LÝ MẪU THIẾT KẾ VÀ
THÀNH PHẦN TỪ MẪU THIẾT KẾ
PHỤC VỤ CHO VIỆC XÂY DỰNG HỆ THỐNG THÔNG TIN**

KHÓA LUẬN CỬ NHÂN TIN HỌC

**GIÁO VIÊN HƯỚNG DẪN
PGS. TS. ĐỒNG THỊ BÍCH THỦY**

TP. HCM, 2005

LỜI CẢM ƠN

Lời đầu tiên chúng em xin chân thành cảm ơn Cô Đồng Thị Bích Thủy, người đã trực tiếp hướng dẫn, góp ý, động viên, và giúp đỡ chúng em trong suốt thời gian thực hiện đề tài Luận Văn. Cô đã truyền đạt cho chúng em những kiến thức về Tin học nói chung, những tri thức chuyên ngành của đề tài nói riêng, giúp chúng em có một nền tảng vững chắc để hoàn thành Luận Văn này.

Chúng em xin chân thành cảm ơn tất cả các Thầy Cô trong Khoa Công Nghệ Thông Tin, những người đã tận tình dạy bảo và giúp đỡ chúng em trong suốt thời gian học đại học. Những kiến thức nhận được từ các Thầy Cô đã giúp chúng em có được một nền tảng vững chắc để thực hiện Luận Văn cũng như có được sự tự tin cần thiết để mạnh dạn tiếp cận các tri thức ngày càng mới trong sự phát triển mạnh mẽ của ngành Công nghệ Thông tin.

Để hoàn thành Luận Văn này, chúng em không thể không nhắc đến sự động viên, chăm sóc của gia đình và những người thân trong suốt quá trình làm Luận Văn. Chính những sự quan tâm đặc biệt này đã tạo cho chúng em có đủ nghị lực để thực hiện tốt luận văn.

Và cuối cùng, xin gửi lời cảm ơn đến tất cả các bạn bè, những người đã sát cánh cùng với chúng tôi trong suốt bốn năm qua, những người đã chia sẻ cùng với tôi những niềm vui cũng như những khó khăn trong học tập và trong cuộc sống.

Tp.Hồ Chí Minh, tháng 7 năm 2005

Nhóm sinh viên thực hiện

Cao Giang Thiên Tài – Tống Nguyễn Quỳnh Trâm

[illegible]

This image shows a blank sheet of handwriting practice paper. It features ten sets of horizontal lines designed for letter formation. Each set includes a solid top line, a dashed middle line, and a solid bottom line. A large, light gray watermark with the text "KHLOACNIT" is placed diagonally across the entire page.

MỤC LỤC

Chương 1	Tổng quan	8
1.1	Dẫn nhập	9
1.1.1	Lý do hình thành mẫu	9
1.1.2	Phân loại mẫu	9
1.1.3	Mẫu thiết kế giao diện	10
1.1.4	Mẫu thiết kế giao diện - ứng dụng Hệ thống thông tin	10
1.2	Bài toán giải quyết	11
1.2.1	Đối tượng sử dụng hệ thống	11
1.2.2	Các yêu cầu đặt ra cho hệ thống	12
1.3	Bố cục của Luận Văn	13
Chương 2	Các mẫu thiết kế giao diện và các mẫu thiết kế	14
2.1	Các khái niệm	15
2.1.1	Loại giao diện	15
2.1.2	Thành phần	16
2.1.3	Mẫu thiết kế giao diện	17
2.1.4	Thể hiện mẫu	20
2.2	Các mẫu giao diện trong hệ thống thông tin	21
2.2.1	Một số thành phần trong hệ thống thông tin	21
2.2.2	Một số loại mẫu thiết kế giao diện trong hệ thống thông tin	32
2.3	Các mẫu thiết kế	48
2.3.1	Mẫu Composite	48
2.3.2	Mẫu Proxy	50
Chương 3	Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng hệ thống thông tin	53
3.1	Phân tích	54
3.1.1	Các sơ đồ Use-Case	54
3.1.2	Phân tích các chức năng chính của hệ thống	63
3.1.3	Sơ đồ lớp mức phân tích	68
3.2	Thiết kế	69
3.2.1	Kiến trúc tổng thể	69
3.2.2	Sơ đồ tuần tự - sơ đồ cộng tác cho một số xử lý chính:	74
3.2.3	Thiết kế lớp	88
3.2.4	Thiết kế Lưu trữ:	97
Chương 4	Cài đặt và thử nghiệm	99
4.1	Môi trường cài đặt	100
4.2	Các chức năng được cài đặt	101
4.2.1	Cây chức năng	101
4.2.2	Quản lý hệ thống	102
4.2.3	Quản lý thông tin cá nhân	103
4.2.4	Quản lý thành phần	104
4.2.5	Quản lý mẫu	108
4.2.6	Quản lý thể hiện mẫu	114
4.2.7	Quản lý module phát sinh mã nguồn	118

4.2.8	Quản lý module nhập từ mã nguồn.....	121
4.3	Thử nghiệm.....	123
4.3.1	Minh họa khả năng tái sử dụng các thành phần của hệ thống	123
4.3.2	Các chức năng Phân quyền, quản lý hệ thống, thông tin cá nhân	124
4.3.3	Quản lý các module nhập vào mã nguồn và phát sinh mã nguồn.....	124
4.3.4	Công cụ đồ họa	124
4.3.5	Quản lý thành phần	125
4.3.6	Quản lý mẫu.....	125
4.3.7	Quản lý thể hiện mẫu	126
Chương 5	Tổng kết và hướng phát triển.....	127
5.1	Kết quả đạt được	128
5.2	Hạn chế	129
5.3	Hướng phát triển	129
Phụ lục	131
	Chi tiết các lớp xử lý chính.....	131
	Chi tiết các bảng trong cơ sở dữ liệu:	149
	Thiết kế chi mục:	151
	Thiết kế các stored procedure:	152
Tài liệu tham khảo	155

DANH MỤC CÁC HÌNH

Hình 2-1 Màn hình nhập khách hàng.....	15
Hình 2-2 Các thành phần của màn hình nhập khách hàng.....	16
Hình 2-3 Mô hình loại mẫu Simple cho giao diện Nhập.....	17
Hình 2-4 Mẫu thiết kế cho giao diện nhập theo loại mẫu Simple	18
Hình 2-5 Màn hình Nhập khách hàng.....	20
Hình 2-6 Bộ lọc có 1 tiêu chí cần được nhập dữ liệu	21
Hình 2-7 Bộ lọc có nhiều tiêu chí cần được nhập dữ liệu	21
Hình 2-8 Bộ lọc có 1 tiêu chí cần được chọn dữ liệu – Dạng 1.....	22
Hình 2-9 Bộ lọc có 1 tiêu chí cần được chọn dữ liệu – Dạng 2.....	22
Hình 2-10 Bộ lọc có nhiều tiêu chí cần được chọn dữ liệu	22
Hình 2-11 Bộ lọc kết hợp các tiêu chí cần nhập và cần chọn dữ liệu – Dạng 1	22
Hình 2-12 Bộ lọc kết hợp các tiêu chí cần nhập và chọn dữ liệu – Dạng 2	23
Hình 2-13 Bộ lọc cho phép kết hợp tất cả hay một vài tiêu chí – Dạng 1	23
Hình 2-14 Bộ lọc cho phép kết hợp tất cả hay một vài tiêu chí – Dạng 2.....	23
Hình 2-15 Bộ lọc dùng cây tra cứu.....	24
Hình 2-16 Bộ hiển thị cho một thông tin.....	24
Hình 2-17 Bộ hiển thị cho nhiều thông tin – Dạng 1.....	25
Hình 2-18 Bộ hiển thị cho nhiều thông tin – Dạng 2.....	25
Hình 2-19 Bộ hiển thị cho nhiều thông tin – Dạng 3.....	25
Hình 2-20 Bộ hiển thị cho nhiều thông tin – Dạng 4.....	25
Hình 2-21 Bộ hiển thị cho nhiều thông tin – Dạng 5.....	25
Hình 2-22 Bộ hiển thị thông tin nhiều đối tượng cùng loại.....	26
Hình 2-23 Bộ hiển thị kết hợp nhóm 1 và nhóm 2 – Dạng 1.....	26
Hình 2-24 Bộ hiển thị kết hợp nhóm 1 và nhóm 2 – Dạng 1.....	27
Hình 2-25 Bộ chỉnh sửa với 1 thông tin cần được nhập	27
Hình 2-26 Bộ chỉnh sửa với 1 thông tin cần được chọn – Dạng 1	27
Hình 2-27 Bộ chỉnh sửa với 1 thông tin cần được chọn – Dạng 2	28
Hình 2-28 Bộ chỉnh sửa với nhiều thông tin cần được nhập	28
Hình 2-29 Bộ chỉnh sửa với nhiều thông tin cần được chọn	28
Hình 2-30 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 1	28
Hình 2-31 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 2.....	28
Hình 2-32 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 3.....	29
Hình 2-33 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 4.....	29
Hình 2-34 Bộ chỉnh sửa thông tin nhiều đối tượng cùng loại.....	29
Hình 2-35 Bộ chỉnh sửa kết hợp nhóm 1 và nhóm 2 – Dạng 1	30
Hình 2-36 Bộ chỉnh sửa kết hợp nhóm 1 và nhóm 2 – Dạng 2	30
Hình 2-37 Bộ các nút chức năng hỗ trợ lưu.....	31
Hình 2-38 Bộ các nút chức năng hỗ trợ tra cứu bằng wizard	31
Hình 2-39 Bộ các nút chức năng hỗ trợ cập nhật	31
Hình 2-40 Bộ các nút chức năng hỗ trợ xóa	31
Hình 2-41 Bộ các nút chức năng hỗ trợ cập nhật và xóa.....	31
Hình 2-42 Bộ các nút chức năng hỗ trợ kết xuất	32
Hình 2-43 Simple – Nhập – Dạng 1	33
Hình 2-44 Mẫu thiết kế giao diện của Simple – Nhập – Dạng 1	33
Hình 2-45 Thẻ hiện mẫu Nhập khách hàng – Dạng 1	34

Hình 2-46 Simple – Nhập – Dạng 2	34
Hình 2-47 Mẫu thiết kế giao diện của Simple – Nhập – Dạng 2.....	35
Hình 2-48 Thể hiện mẫu Nhập khách hàng – Dạng 2	35
Hình 2-49 Simple – Tra cứu	36
Hình 2-50 Mẫu thiết kế giao diện của Simple – Tra cứu.....	37
Hình 2-51 Thể hiện mẫu Tra cứu khách hàng	37
Hình 2-52 Simple – Chỉnh sửa	38
Hình 2-53 Mẫu thiết kế giao diện của Simple – Chỉnh sửa.....	38
Hình 2-54 Thể hiện mẫu Chỉnh sửa thông tin khách hàng	39
Hình 2-55 Simple – Xuất	39
Hình 2-56 Mẫu thiết kế giao diện của Simple – Xuất	40
Hình 2-57 Thể hiện mẫu Xuất thông tin khách hàng.....	40
Hình 2-58 Master Detail – Nhập.....	41
Hình 2-59 Mẫu thiết kế giao diện của Master Detail – Nhập.....	42
Hình 2-60 Thể hiện mẫu Nhập Danh sách nhân viên.....	42
Hình 2-61 Master Detail – Tra cứu.....	43
Hình 2-62 Mẫu thiết kế giao diện của Master Detail – Tra cứu	43
Hình 2-63 Thể hiện mẫu Tra cứu Nhân viên	44
Hình 2-64 Master Detail – Chỉnh sửa.....	44
Hình 2-65 Mẫu thiết kế giao diện của Master Detail – Chỉnh sửa	45
Hình 2-66 Thể hiện mẫu Chỉnh sửa thông tin nhân viên.....	45
Hình 2-67 Master Detail – Xuất	46
Hình 2-68 Mẫu thiết kế giao diện của Master Detail – Xuất.....	46
Hình 2-69 Thể hiện mẫu Xuất danh sách nhân viên.....	47
Hình 2-70 Sơ đồ UML cho mẫu Composite.....	48
Hình 2-71 Mô hình triển khai mẫu composite trong hệ thống.....	50
Hình 2-72 Sơ đồ UML mẫu Proxy	51
Hình 2-73 Mô hình mẫu Proxy tại thời điểm thực thi	51
Hình 2-74 Mô hình thể hiện ý tưởng thiết kế theo mẫu Proxy	52
Hình 3-1 Sơ đồ Use-Case của toàn hệ thống	54
Hình 3-2 Sơ đồ Use-Case Quản lý hệ thống.....	56
Hình 3-3 Sơ đồ Use-Case Quản lý thông tin cá nhân.....	57
Hình 3-4 Sơ đồ Use-Case Quản lý thành phần.....	58
Hình 3-5 Sơ đồ Use-Case Quản lý mẫu.....	59
Hình 3-6 Sơ đồ Use-Case Quản lý thể hiện mẫu	60
Hình 3-7 Sơ đồ Use-Case Quản lý module nhập vào mã nguồn	61
Hình 3-8 Sơ đồ Use-Case Quản lý module phát sinh mã nguồn	62
Hình 3-9 Sơ đồ lớp mức phân tích.....	68
Hình 3-10 Kiến trúc triển khai.....	69
Hình 3-11 Mô hình ba tầng được triển khai	70
Hình 3-12 Kiến trúc logic trên hệ thống chính	71
Hình 3-13 Mô tả chi tiết nhóm các lớp Điều Khiển	72
Hình 3-14 Kiến trúc logic trên các module nhập và các module phát sinh mã nguồn	73
Hình 3-15 Sơ đồ tuần tự của xử lý tra cứu thành phần.....	74
Hình 3-16 Sơ đồ tuần tự của xử lý chi tiết tra cứu thành phần.....	75
Hình 3-17 Sơ đồ cộng tác xử lý tra cứu thành phần	76
Hình 3-18 Sơ đồ cộng tác chi tiết xử lý tra cứu thành phần	76

Hình 3-19 Sơ đồ tuần tự xử lý cập nhật thành phần	77
Hình 3-20 Sơ đồ tuần tự chi tiết cập nhật thành phần	78
Hình 3-21 Sơ đồ cộng tác xử lý cập nhật thành phần	79
Hình 3-22 Sơ đồ cộng tác chi tiết xử lý cập nhật thành phần	79
Hình 3-23 Sơ đồ tuần tự xử lý phát sinh mã nguồn gọi từ menu chính.....	80
Hình 3-24 Sơ đồ tuần tự chi tiết xử lý phát sinh mã nguồn cho thể hiện mẫu	81
Hình 3-25 Sơ đồ cộng tác xử lý phát sinh mã nguồn cho thể hiện mẫu gọi từ menu chính	82
Hình 3-26 Sơ đồ cộng tác chi tiết xử lý phát sinh mã nguồn trong module	82
Hình 3-27 Sơ đồ tuần tự cho xử lý phát sinh mã nguồn cho thể hiện mẫu đang thiết kế....	83
Hình 3-28 Sơ đồ cộng tác xử lý phát sinh mã nguồn cho thể hiện mẫu đang thiết kế.....	84
Hình 3-29 Sơ đồ tuần tự xử lý nhập mã nguồn cho mẫu	85
Hình 3-30 Sơ đồ tuần tự chi tiết xử lý nhập mã nguồn trong module	86
Hình 3-31 Sơ đồ cộng tác xử lý nhập mã nguồn cho Mẫu	87
Hình 3-32 Sơ đồ cộng tác chi tiết xử lý nhập mã nguồn	88
Hình 3-33 Sơ đồ lớp của nhóm các điều khiển.....	89
Hình 3-34 Sơ đồ lớp phục vụ chức năng quản lý thành phần.....	90
Hình 3-35 Sơ đồ lớp phục vụ chức năng quản lý mẫu	91
Hình 3-36 Sơ đồ lớp phục vụ chức năng quản lý thành phần.....	92
Hình 3-37 Sơ đồ lớp phục vụ chức năng quản lý module nhập vào mã nguồn.....	93
Hình 3-38 Sơ đồ lớp phục vụ chức năng quản lý module phát sinh mã nguồn.....	94
Hình 3-39 Sơ đồ lớp phục vụ chức năng phát sinh mã nguồn.....	95
Hình 3-40 Sơ đồ lớp phục vụ chức năng nhập vào mã nguồn.....	96
Hình 3-41 Cấu trúc tổ chức thư mục và tập tin lưu hình ảnh minh họa	97
Hình 3-42 Sử dụng Xml trong cơ sở dữ liệu QLMauGiaoDien	97
Hình 3-43 Sơ đồ logic.....	98
Hình 4-1 Màn hình cho phép thêm một người dùng mới	102
Hình 4-2 Màn hình xem thông tin cá nhân	103
Hình 4-3 Màn hình hiệu chỉnh thông tin cá nhân	103
Hình 4-4 Màn hình cho phép thiết kế và quản lý thành phần.....	104
Hình 4-5 Wizard cho phép tra cứu thành phần	105
Hình 4-6 Màn hình mở đặc tả xml cho thành phần	106
Hình 4-7 Màn hình quản lý mẫu	108
Hình 4-8 Tra cứu mẫu.....	109
Hình 4-9 Màn hình xóa mẫu	111
Hình 4-10 Màn hình nhập mã nguồn	112
Hình 4-11 Màn hình phát sinh mã nguồn	113
Hình 4-12 Màn hình quản lý Thể hiện mẫu.....	114
Hình 4-13 Màn hình tra cứu thể hiện mẫu.....	115
Hình 4-14 Màn hình xóa thể hiện mẫu	116
Hình 4-15 Màn hình thêm module phát sinh mã nguồn	118
Hình 4-16 Màn hình xóa module phát sinh mã nguồn	119
Hình 4-17 Màn hình cập nhật module phát sinh mã nguồn.....	120
Hình 4-18 Màn hình thêm module nhập mã nguồn	121
Hình 4-19 Màn hình cập nhật module nhập mã nguồn.....	122

DANH MỤC CÁC BẢNG

Bảng 3-1 Bảng thuyết minh Sơ đồ Use-Case của toàn hệ thống.....	55
Bảng 3-2 Bảng thuyết minh Sơ đồ Use-Case Quản lý hệ thống.....	56
Bảng 3-3 Bảng thuyết minh Sơ đồ Use-Case Quản lý thông tin cá nhân.....	57
Bảng 3-4 Bảng thuyết minh Sơ đồ Use-Case Quản lý thành phần.....	58
Bảng 3-5 Bảng thuyết minh Sơ đồ Use-Case Quản lý mẫu.....	59
Bảng 3-6 Bảng thuyết minh Sơ đồ Use-Case Quản lý thể hiện mẫu.....	60
Bảng 3-7 Bảng thuyết minh Sơ đồ Use-Case Quản lý module nhập khẩu mã nguồn.....	61
Bảng 3-8 Bảng thuyết minh Sơ đồ Use-Case Quản lý module phát sinh mã nguồn.....	62
Bảng 3-9 Danh sách các điều khiển chuẩn trong ngôn ngữ lập trình C# và VB 6.0.....	64
Bảng 3-10 Danh sách lớp phục vụ chức năng quản lý.....	89
Bảng 3-11 Danh sách lớp phục vụ chức năng quản lý thành phần.....	90
Bảng 3-12 Danh sách lớp phục vụ chức năng quản lý mẫu.....	91
Bảng 3-13 Danh sách lớp phục vụ chức năng quản lý thành phần.....	92
Bảng 3-14 Danh sách lớp phục vụ chức năng quản lý module nhập vào mã nguồn.....	93
Bảng 3-15 Danh sách lớp phục vụ chức năng quản lý module phát sinh mã nguồn.....	94
Bảng 3-16 Danh sách lớp phục vụ chức năng phát sinh mã nguồn.....	95
Bảng 3-17 Danh sách lớp phục vụ chức năng phát sinh mã nguồn.....	96
Bảng 3-18 Danh sách các bảng quan hệ trong cơ sở dữ liệu QLMauGiaoDien.....	98
Bảng PL - 1 Danh sách thuộc tính lớp ThanhPhan.....	132
Bảng PL - 2 Danh sách hàm lớp ThanhPhan.....	133
Bảng PL - 3 Danh sách thuộc tính lớp XL_ThanhPhan.....	134
Bảng PL - 4 Danh sách hàm lớp XL_ThanhPhan.....	134
Bảng PL - 5 Danh sách hàm lớp LT_ThanhPhan.....	135
Bảng PL - 6 Danh sách thuộc tính lớp Mau.....	136
Bảng PL - 7 Danh sách hàm lớp Mau.....	137
Bảng PL - 8 Danh sách thuộc tính lớp XL_Mau.....	138
Bảng PL - 9 Danh sách hàm lớp XL_Mau.....	139
Bảng PL - 10 Danh sách hàm lớp LT_Mau.....	139
Bảng PL - 11 Danh sách thuộc tính lớp TheHienMau.....	141
Bảng PL - 12 Danh sách hàm lớp TheHienMau.....	142
Bảng PL - 13 Danh sách thuộc tính lớp XL_TheHienMau.....	143
Bảng PL - 14 Danh sách hàm XL_TheHienMau.....	143
Bảng PL - 15 Danh sách thuộc tính lớp LT_TheHienMau.....	144
Bảng PL - 16 Danh sách các thuộc tính lớp MyControl.....	145
Bảng PL - 17 Danh sách hàm lớp MyControl.....	146
Bảng PL - 18 Danh sách hàm lớp MyLabel.....	147
Bảng PL - 19 Danh sách thuộc tính lớp MyComposite.....	148
Bảng PL - 20 Danh sách hàm lớp MyComposite.....	148
Bảng PL - 21 Chi tiết bảng Quyen.....	149
Bảng PL - 22 Chi tiết bảng NguoiDung.....	149
Bảng PL - 23 Chi tiết bảng LoaiMau.....	149
Bảng PL - 24 Chi tiết bảng NguoiDung.....	149
Bảng PL - 25 Chi tiết bảng ThanhPhan.....	149

Bảng PL - 26 Chi tiết bảng Mẫu	150
Bảng PL - 27 Chi tiết bảng Thể hiện mẫu	150
Bảng PL - 28 Chi tiết bảng NgonNguPhatSinh	151
Bảng PL - 29 Chi tiết bảng NgonNguNhapMoi	151
Bảng PL - 30 Danh sách chỉ mục được thiết kế	151
Bảng PL - 31 Danh sách các store procedure được thiết kế	154

KHOA CNTT

Chương 1 Tổng quan

Nội dung

Chương này sẽ giới thiệu tổng quan về sự hình thành mẫu, phân loại mẫu, phân tích tầm quan trọng và ý nghĩa thực tiễn của mẫu thiết kế giao diện trong việc phát triển các ứng dụng Hệ thống thông tin. Từ đó, đưa ra mục đích và ý nghĩa của đề tài xây dựng hệ thống quản lý mẫu thiết kế giao diện.

KHOA CNTT

1.1 Dẫn nhập

1.1.1 Lý do hình thành mẫu

Một trong những tiêu chí quan trọng để đánh giá chất lượng phần mềm là khả năng tái sử dụng. Khảo sát cho thấy mỗi đề án phần mềm là một thực thể phức tạp và duy nhất, nhưng giữa các thực thể đó, ta vẫn tìm thấy sự lặp lại ở một hoặc nhiều điểm. Qua quá trình nghiên cứu, hướng giải quyết cho các vấn đề mang tính chất lặp đi lặp lại khi phát triển các hệ thống phần mềm đã được các chuyên gia đúc kết, tinh chỉnh và nâng lên một mức tổng quát. Cũng từ đó khái niệm “mẫu” ra đời. Có nhiều định nghĩa khác nhau về “mẫu” như sau:

Christophter Alexander đã định nghĩa *“Mỗi mẫu mô tả một vấn đề mang tính chất xuất hiện lặp đi lặp lại trong môi trường của chúng ta, một giải pháp cho vấn đề đó”*[1]

Một định nghĩa khác về mẫu của Gamma et. al: *“Một mẫu thể hiện bản chất cốt lõi của một giải pháp cho một vấn đề xuất hiện lặp đi lặp lại trong một ngữ cảnh được chỉ định rõ”*.

Trong khi đó Fowler có một định nghĩa khác: *“Mẫu là một ý tưởng đã được sử dụng trong một ngữ cảnh thực tế và sẽ được áp dụng một cách chính xác cho các trường hợp khác”*.

Người phát triển phần mềm sẽ cụ thể hóa những giải pháp mà “mẫu” mang lại để giải quyết những bài toán trong thực tế mà họ cảm thấy phù hợp, đó gọi là sử dụng “mẫu”.

1.1.2 Phân loại mẫu

Mẫu được phân thành hai loại chính sau: mẫu phân tích và mẫu thiết kế. Hai loại mẫu này phục vụ đặc lực cho hai pha phân tích, thiết kế của qui trình phát triển phần mềm.

[1] Nguyễn Trần Minh Thư. Báo cáo Luận Văn Cao Học. Trường Đại học Khoa học Tự Nhiên. Năm 2004.

Mẫu phân tích là các mô hình quan niệm đã được xây dựng để mô hình hóa tri thức cốt lõi của một vấn đề. Vì thế có sự mong đợi rằng các mẫu đã sử dụng để mô hình hóa cho một vấn đề cụ thể được sử dụng lại để mô hình hóa cho vấn đề tương tự một cách dễ dàng và thành công bất chấp phạm vi ngữ cảnh của vấn đề xuất hiện.

Mẫu thiết kế mô tả có hệ thống một giải pháp đã được kiểm nghiệm cho một vấn đề thiết kế xuất hiện nhiều lần như: tạo lập lớp/đối tượng, thiết kế cấu trúc lớp/đối tượng và các hành vi của lớp/đối tượng. Một mẫu thiết kế cũng mô tả những khả năng, sự cân bằng các yếu tố để đạt được sự kết hợp tốt nhất, và các hệ quả của giải pháp. Nó cũng có thể minh họa làm thế nào để thực hiện một giải pháp trong ngôn ngữ lập trình chẳng hạn như C, C++, C#, VB.net, Java,...[Gamma et al.1995]. Vì thế nó cung cấp cho các nhà phát triển các phương pháp thiết kế đã được chứng minh và các hướng dẫn để thực hiện chúng.

1.1.3 Mẫu thiết kế giao diện

Mẫu thiết kế là một khái niệm rộng và bao quát trong công đoạn thiết kế phần mềm: thiết kế đối tượng, thiết kế giao diện... Vì thế ngoài những mẫu thiết kế lớp/đối tượng, còn có các mẫu hướng dẫn thiết kế giao diện.

Trong quá trình phát triển ứng dụng, việc thiết kế giao diện được lặp đi lặp lại và nhận thấy rằng có rất nhiều giao diện tương tự nhau về chủng loại. Từ nhận định đó, các chuyên gia đã thu thập và tổng hợp giao diện thường gặp trong hầu hết các ứng dụng để đưa ra mẫu thiết kế giao diện. Thiết kế giao diện tuân theo mẫu sẽ trở nên đơn giản và hiệu quả hơn vì giao diện được thiết kế sẽ đã đảm bảo những tiêu chuẩn cần có mà mẫu thiết kế giao diện đề ra.

1.1.4 Mẫu thiết kế giao diện - ứng dụng Hệ thống thông tin

Vai trò của mẫu thiết kế giao diện thể hiện rõ hơn trong quá trình xây dựng các ứng dụng Hệ thống thông tin. Số lượng màn hình giao diện trong một ứng dụng thuộc lĩnh vực này thường không nhỏ và có sự lặp lại ở tần suất cao về chủng loại, đơn cử như: bất kỳ hệ thống thông tin nào cũng cần những màn hình nhập liệu, màn

hình báo biểu,... Vì vậy áp dụng mẫu thiết kế giao diện vào việc xây dựng ứng dụng Hệ thống thông tin là một việc làm đem lại nhiều lợi ích.

Để đưa mẫu thiết kế giao diện vào công việc chuyên môn của những người phát triển ứng dụng Hệ thống thông tin một cách tự nhiên nhất, chúng tôi quyết định xây dựng một **Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng Hệ thống thông tin** với những mục đích và ý nghĩa sau:

- Hệ thống sẽ quản lý những mẫu thiết kế giao diện do các nhà nghiên cứu mẫu cung cấp, từ đó làm cơ sở giúp người sử dụng có thể thiết kế giao diện nhanh chóng và hiệu quả.
- Hệ thống cũng quản lý các thành phần từ mẫu thiết kế giao diện để tăng cường khả năng tái sử dụng. Người dùng không những có thể sử dụng lại toàn bộ mẫu thiết kế giao diện mà còn có thể sử dụng từng thành phần thích hợp cho giao diện của mình.
- Ngoài ra hệ thống còn quản lý những thể hiện của mẫu thiết kế giao diện từ các ứng dụng mà người sử dụng đang phát triển để giúp họ có thể tái sử dụng những giao diện này khi cần thiết.

Đó cũng chính là đề tài Luận Văn tốt nghiệp mà chúng tôi lựa chọn.

1.2 Bài toán giải quyết

Bài toán đặt ra đó là xây dựng **Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng Hệ thống thông tin** phục vụ cho việc xây dựng Hệ thống thông tin nhằm xoáy mạnh vào khả năng tái sử dụng các mẫu thiết kế giao diện.

1.2.1 Đối tượng sử dụng hệ thống

- **Nhà nghiên cứu mẫu** là những người nghiên cứu tìm ra các mẫu thiết kế giao diện. Nhà nghiên cứu mẫu cần sử dụng hệ thống như một công cụ lưu trữ những kết quả làm việc. Hệ thống phải hỗ trợ nhà nghiên cứu thực hiện công việc chuyên môn một cách hiệu quả nhất.

- **Người sử dụng mẫu:** là những người phát triển các ứng dụng, đặc biệt là các ứng dụng Hệ thống thông tin. Họ sẽ sử dụng hệ thống để được hướng dẫn cho việc thiết kế giao diện, tái sử dụng những giao diện sẵn có, đồng thời lưu trữ lại kết quả thiết kế của mình.

1.2.2 Các yêu cầu đặt ra cho hệ thống

Để có thể đáp ứng tốt những yêu cầu của các đối tượng sử dụng, **Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng Hệ thống thông tin** cần đáp ứng những yêu cầu sau:

- Có cách đặc tả hiệu quả cho giao diện vì đây là một kiểu dữ liệu đặc biệt.
- Hệ thống phải quản lý màn hình giao diện theo thành phần. Nhờ vậy, người dùng có thể tái sử dụng toàn bộ màn hình hay một thành phần bộ phận.
- Xây dựng công cụ đồ họa để hỗ trợ người dùng thiết kế mẫu giao diện.
- Hệ thống hỗ trợ chức năng nhập một màn hình giao diện vào hệ thống từ tập tin mã nguồn .
- Hệ thống phải hỗ trợ các chức năng thêm, cập nhật và xóa các đối tượng được quản lý.
- Hỗ trợ tra cứu sinh động và hiệu quả thông qua wizard.
- Để người dùng có thể sử dụng ngay màn hình giao diện mà hệ thống trả về cần phải xây dựng chức năng phát sinh mã nguồn cho màn hình giao diện mà hệ thống lưu trữ.
- Bài toán quản lý mẫu giao diện là một bài toán rất động có khả năng biến đổi nhiều trong tương lai nên hệ thống sẽ được phân tích thiết kế hướng đối tượng và cài đặt hoàn toàn theo mô hình ba tầng.
- Các chức năng phát sinh mã nguồn và nhập màn hình giao diện từ tập tin mã nguồn sẽ được module hóa để có thể mở rộng trong tương lai.

1.3 Bố cục của Luận Văn

Chương 1: Tổng quan.

Chương 2: Các mẫu thiết kế giao diện và các mẫu thiết kế

Chương 3: Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng hệ thống thông tin.

Chương 4: Cài đặt và thử nghiệm.

Chương 5: Kết luận và hướng phát triển.

Phụ lục.

Tài liệu tham khảo.

KHOA CNTT

Chương 2 Các mẫu thiết kế giao diện và các mẫu thiết kế

Nội dung

Chương này sẽ bắt đầu bằng việc làm rõ các khái niệm liên quan đến mẫu thiết kế giao diện. Tiếp theo là phần hệ thống hóa một số mẫu thiết kế giao diện trong hệ thống thông tin. Cuối cùng là phần trình bày về hai mẫu thiết kế Composite và Proxy (trong số 23 mẫu GoF) được ứng dụng để xây dựng Hệ thống.

KHOA CNTT

2.1 Các khái niệm

2.1.1 Loại giao diện

Loại giao diện là một khái niệm nhằm phân loại các màn hình giao diện dựa vào chức năng của nó.

Trong hệ thống thông tin, có 4 loại giao diện được sử dụng phổ biến: loại giao diện Nhập, loại giao diện Tra cứu, loại giao diện Chính sửa (cập nhật, xóa) và loại giao diện Xuất (xem thông tin đơn giản, làm báo biểu...).

Ví dụ minh họa: màn hình sau thuộc loại giao diện Nhập

Tiêu đề màn hình

NHẬP KHÁCH HÀNG

Tên khách hàng

Loại khách hàng

Địa chỉ

Hình 2-1 Màn hình nhập khách hàng

Với hệ thống quản lý mẫu giao diện theo hướng thành phần, sử dụng khái niệm loại giao diện sẽ giúp cho việc quản lý các màn hình theo chức năng được tốt hơn, mở rộng thêm tiêu chí tra cứu cho người sử dụng. Đây cũng là cơ sở để thống kê tần suất sử dụng các loại giao diện trong các hệ thống thông tin về sau.

2.1.2 Thành phần

Thành phần là một nhóm các điều khiển đảm nhận một chức năng nhất định trong một màn hình giao diện.

Ví dụ minh họa:

The image shows a screenshot of a web application form titled 'NHẬP KHÁCH HÀNG' (Enter Customer). The form is enclosed in a blue border. At the top, there is a light blue header bar with the text 'Tiêu đề màn hình' (Form Title) and 'NHẬP KHÁCH HÀNG'. Below this, there is a green rectangular area containing three input fields: 'Tên khách hàng' (Customer Name) with the value 'Nguyễn Văn Anh', 'Loại khách hàng' (Customer Type) with a dropdown menu showing 'Khách hàng thân thiết' (Loyal Customer), and 'Địa chỉ' (Address) with the value '85/12 Nguyễn Kiệm Phú Nhuận TP.HCM'. At the bottom of the form, there is an orange bar containing three buttons: 'Lưu' (Save), 'Bỏ qua' (Skip), and 'Thoát' (Exit). An orange box labeled 'Các thành phần' (Components) has three arrows pointing to the green area, the orange bar, and the blue border, indicating that these are the components of the form.

Hình 2-2 Các thành phần của màn hình nhập khách hàng

Vì sao lại có khái niệm thành phần? Nhu cầu này bắt nguồn từ việc thiết kế các màn hình giao diện. Khi thiết kế, ai trong chúng ta cũng mong muốn giao diện của mình không những hoạt động một cách đúng đắn, đầy đủ các chức năng của nó mà còn mong muốn rằng, nó mang lại cho người dùng một cảm giác thân thiện và tiện dụng. Để đạt được điều này, ta cần xây dựng tốt bố cục của màn hình, nói cách khác là ta phải bố trí các điều khiển sao cho mọi hoạt động diễn ra một cách liền mạch và phù hợp với thói quen của người dùng. Để việc xây dựng bố cục được thực hiện dễ dàng, ta sẽ nhóm các điều khiển cùng đảm nhận một chức năng nào đó trong màn hình lại với nhau. Nhóm các điều khiển như vậy được gọi là thành phần.

Sự ra đời của khái niệm thành phần còn nhằm phục vụ cho một mục tiêu quan trọng hơn, đó là khả năng tái sử dụng khi thiết kế giao diện. Như đã trình bày ở chương 1, khi thiết kế giao diện trong các hệ thống thông tin, người ta nhận ra rằng có rất nhiều màn hình giống nhau về cách thức hoạt động, nếu ta gỡ bỏ ngữ cảnh sử dụng của các màn hình này thì thực chất chúng chỉ là một! Hay lại có

những màn hình với cách thức hoạt động chỉ khác nhau ở một công đoạn nào đó, còn các công đoạn khác lại hoàn toàn giống nhau. Thành phần hóa các màn hình giao diện là cách tốt nhất để ta có thể tận dụng những gì đã thực hiện trước đó, thay vì phải làm lại từ đầu với cảm giác nhàm chán!

*Trong hệ thống quản lý mẫu giao diện theo hướng thành phần, việc thiết kế một màn hình giao diện là **sự lắp ghép các thành phần một cách hợp lý** để thực hiện các yêu cầu chức năng và phi chức năng của màn hình.*

2.1.3 Mẫu thiết kế giao diện

Loại mẫu thiết kế giao diện là một tập hợp các mẫu thiết kế giao diện có cùng một kết cấu hình thành.

Ví dụ minh họa:



Hình 2-3 Mô hình loại mẫu Simple cho giao diện Nhập

Mẫu thiết kế giao diện là một giao diện được xây dựng dựa trên loại mẫu thiết kế giao diện ở mức độ trừu tượng hóa, chưa có ngữ cảnh sử dụng cụ thể.

Ví dụ minh họa:



Hình 2-4 Mẫu thiết kế cho giao diện nhập theo loại mẫu Simple

Thông thường khi thiết kế một màn hình trong hệ thống thông tin, ta sẽ thiết kế cho phù hợp với ngữ cảnh sử dụng, nghĩa là các điều khiển đã có những giá trị ấn định cụ thể. Nhưng khi thiết kế trên nhiều hệ thống và bắt gặp sự trùng lặp, ta sẽ nảy sinh ý tưởng phải làm thế nào để có thể tái sử dụng chúng. Muốn vậy, ta phải trừu tượng hóa chúng lên, gỡ bỏ ngữ cảnh đang làm bó hẹp phạm vi sử dụng của chúng. Hệ thống quản lý mẫu giao diện theo hướng thành phần sẽ tiếp nhận chúng như những khuôn mẫu để từ đó, người dùng chỉ cần áp đặt thêm ngữ cảnh sử dụng cụ thể, chỉnh chu đôi chút là có ngay một giao diện ưng ý.

Khi đã có những mẫu thiết kế giao diện và đưa vào sử dụng một thời gian, ta lại phát hiện ra rằng, nếu chỉnh sửa một số thành phần (thêm, bớt, thay đổi bố cục) thì mẫu thiết kế giao diện ban đầu có thể đáp ứng được yêu cầu của một loạt các giao diện cụ thể trong hệ thống thông tin. Vấn đề đặt ra là ta sẽ nhìn nhận thế nào về

mẫu thiết kế giao diện ban đầu (gọi là mẫu A) và mẫu giao diện thiết kế được chỉnh sửa về sau (gọi là mẫu B). Nếu như việc chỉnh sửa này chỉ ở mức thấp hay tương đối, B vẫn còn đó cái hồn của A, ta cần xem nó là một biến thể của A. Nhưng nếu quá trình chỉnh sửa lại tạo ra một mẫu B với một phong cách khác, đến mức B có thể xem là một mẫu độc lập, ngang hàng với A, ta không nên xem nó là biến thể của A nữa.

Khi hệ thống được đưa vào sử dụng một thời gian dài, quá trình tạo lập B từ A sẽ tạo ra một hệ thống các mẫu thiết kế giao diện được phân cấp theo mối quan hệ gốc-biến thể. Cấu trúc phân cấp như vậy sẽ góp phần phân loại các mẫu thiết kế giao diện một cách chặt chẽ hơn, làm cho qui trình tra cứu các mẫu thêm tiện lợi. Bên cạnh đó, nếu có thêm một vài thông số thống kê, ta có thể đưa ra một số khuyến nghị cho việc thiết kế giao diện trong hệ thống thông tin. Chẳng hạn nếu có thêm thông số về số lần được sử dụng của một mẫu thiết kế giao diện, ta có thể phân ra được nhóm những mẫu được dùng nhiều nhất và nhóm những mẫu được dùng ít nhất. Tiến hành phân tích đặc điểm chung của mỗi nhóm, dựa trên kết quả đó, ta sẽ đưa ra khuyến nghị về những điều nên và không nên khi thiết kế giao diện.

2.1.4 Thể hiện mẫu

Thể hiện mẫu là một giao diện được thiết kế gắn liền với một ngữ cảnh sử dụng trong một ứng dụng thực tế, được cụ thể hóa từ mẫu thiết kế giao diện.

Ví dụ minh họa:

The image shows a web form titled "Tiêu đề màn hình" (Screen Title) with the main heading "NHẬP KHÁCH HÀNG" (Enter Customer). The form contains three input fields: "Tên khách hàng" (Customer Name) with the value "Nguyễn Văn Anh", "Loại khách hàng" (Customer Type) with a dropdown menu showing "Khách hàng thân thiết" (Loyal Customer), and "Địa chỉ" (Address) with the value "85/12 Nguyễn Kiệm Phú Nhuận TP.HCM". At the bottom, there are three buttons: "Lưu" (Save), "Bỏ qua" (Skip), and "Thoát" (Exit).

Hình 2-5 Màn hình Nhập khách hàng

Nếu như việc gỡ bỏ ngữ cảnh là cần thiết khi tạo lập một mẫu thiết kế giao diện thì việc hiện thực hóa một mẫu thiết kế giao diện thành một thể hiện mẫu lại cần đến việc xác định ngữ cảnh, ấn định các giá trị cụ thể cho các điều khiển. Thể hiện mẫu sẽ gắn gũi hơn với các ứng dụng cụ thể bên ngoài. Khi việc quản lý các thể hiện mẫu được tổ chức tốt, điều mà người dùng cần làm khi thiết kế giao diện là tra cứu thể hiện mẫu trong hệ thống quản lý mẫu giao diện theo hướng thành phần và phát sinh mã nguồn rồi gắn vào ứng dụng đang cần; không phải đi thiết kế lại từ đầu.

2.2 Các mẫu giao diện trong hệ thống thông tin

2.2.1 Một số thành phần trong hệ thống thông tin

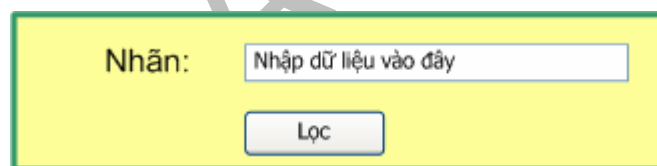
Như khái niệm đã trình bày, thành phần là một nhóm các điều khiển đảm nhận một chức năng nhất định trong một màn hình giao diện. Qua khảo sát và tìm hiểu các giao diện trong các hệ thống thông tin [2, 3], chúng tôi đã đúc kết được một số thành phần được sử dụng phổ biến trong các giao diện này. Cụ thể như sau:

2.2.1.1 Thành phần Bộ lọc

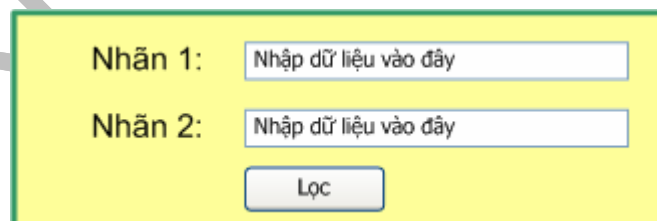
Một bộ lọc là một phương tiện hữu ích cho việc tìm kiếm các đối tượng. Người dùng đưa ra các tiêu chí phù hợp để tìm kiếm được đối tượng mong muốn. Cách bố trí các tiêu chí trong bộ lọc theo một thứ tự đôi khi cần phải tuân thủ để đạt được kết quả chính xác và nhanh chóng. Các nhà phát triển nhận ra rằng *người dùng không thật sự cần một bộ tìm kiếm mạnh, thay vào đó là những bộ lọc được thiết kế dành riêng sẽ gia tăng tính tiện dụng của sản phẩm*. [4]

Một số loại bộ lọc được khảo sát bao gồm

Nhóm bộ lọc 1: các tiêu chí cần được nhập dữ liệu.



Hình 2-6 Bộ lọc có 1 tiêu chí cần được nhập dữ liệu



Hình 2-7 Bộ lọc có nhiều tiêu chí cần được nhập dữ liệu

[2] Scott W. Ambler, userInterfaceDesign, <http://www.ambysoft.com>

[3] Tidwell J., Common Ground: A Pattern Language for Human-Computer Interface Design (1999).

[4] Laakso S.A, User Interfaces, Lecture notes of the 581391-1 User Interfaces course, series D405, Department of Computer Science, University of Helsinki, 2000.

Nhóm bộ lọc 2: các tiêu chí cần được chọn trong một danh sách cho trước

Form này có nền màu vàng. Ở phía trên bên trái có nhãn "Nhãn:". Ngay bên phải nhãn là một menu thả xuống với nội dung "Dữ liệu" và một mũi tên hướng xuống. Dưới menu thả xuống là một nút bấm có chữ "Lọc".

Hình 2-8 Bộ lọc có 1 tiêu chí cần được chọn dữ liệu – Dạng 1

Form này có nền màu vàng. Ở phía trên bên trái có nhãn "Thông tin:". Bên phải nhãn là một khung chứa menu thả xuống với nội dung "Dữ liệu" và các nút điều hướng (lên, xuống, tìm kiếm, v.v.). Dưới khung menu thả xuống là một nút bấm có chữ "Lọc".

Hình 2-9 Bộ lọc có 1 tiêu chí cần được chọn dữ liệu – Dạng 2

Form này có nền màu vàng. Nó chứa hai nhãn "Nhãn 1:" và "Nhãn 2:" xếp chồng lên nhau. Bên phải mỗi nhãn là một menu thả xuống với nội dung "Dữ liệu" và một mũi tên hướng xuống. Dưới hai menu thả xuống là một nút bấm có chữ "Lọc".

Hình 2-10 Bộ lọc có nhiều tiêu chí cần được chọn dữ liệu

Nhóm bộ lọc 3: Trong các tiêu chí tra cứu, có những tiêu chí phải nhập dữ liệu, có những tiêu chí cần được chọn trong một danh sách cho trước. Đây sự kết hợp của bộ lọc 1 và bộ lọc 2.

Form này có nền màu vàng. Nó chứa ba nhãn "Nhãn 1:", "Nhãn 2:" và "Nhãn 3:" xếp chồng lên nhau. Bên phải nhãn "Nhãn 1:" là một trường nhập liệu với nội dung "Nhập dữ liệu vào đây". Bên phải nhãn "Nhãn 2:" và "Nhãn 3:" là hai menu thả xuống với nội dung "Dữ liệu" và một mũi tên hướng xuống. Dưới hai menu thả xuống là một nút bấm có chữ "Lọc".

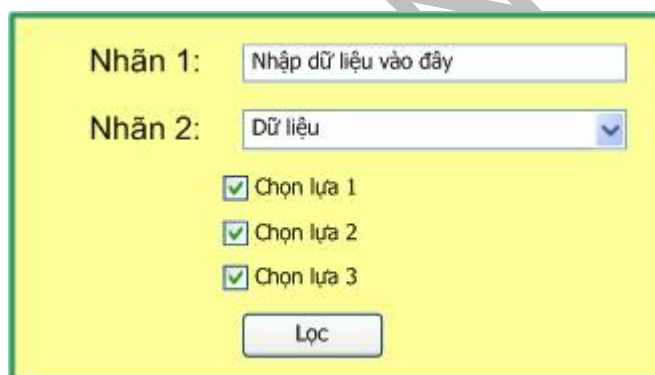
Hình 2-11 Bộ lọc kết hợp các tiêu chí cần nhập và cần chọn dữ liệu – Dạng 1



The form has a yellow background and a green border. It contains two labels: 'Nhãn:' and 'Thông tin:'. The 'Nhãn:' label is followed by a text input field with the placeholder text 'Nhập dữ liệu vào đây'. The 'Thông tin:' label is followed by a large text area with the placeholder text 'Dữ liệu'. At the bottom of the form is a button labeled 'Lọc'.

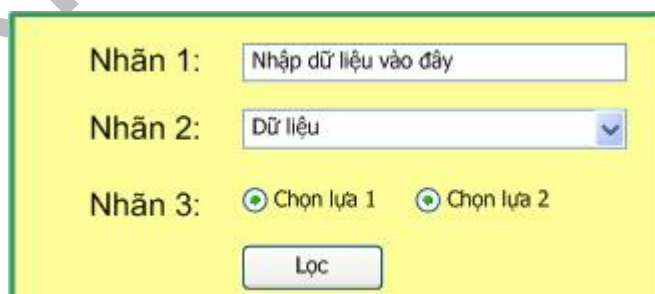
Hình 2-12 Bộ lọc kết hợp hợp các tiêu chí cần nhập và chọn dữ liệu – Dạng 2

Nhóm bộ lọc 4: Trong các tiêu chí tra cứu, ta có thể sử dụng kết hợp tất cả hoặc một vài tiêu chí



The form has a yellow background and a green border. It contains two labels: 'Nhãn 1:' and 'Nhãn 2:'. The 'Nhãn 1:' label is followed by a text input field with the placeholder text 'Nhập dữ liệu vào đây'. The 'Nhãn 2:' label is followed by a dropdown menu with the placeholder text 'Dữ liệu'. Below the dropdown menu are three checkboxes, each followed by a label: 'Chọn lựa 1', 'Chọn lựa 2', and 'Chọn lựa 3'. All three checkboxes are checked. At the bottom of the form is a button labeled 'Lọc'.

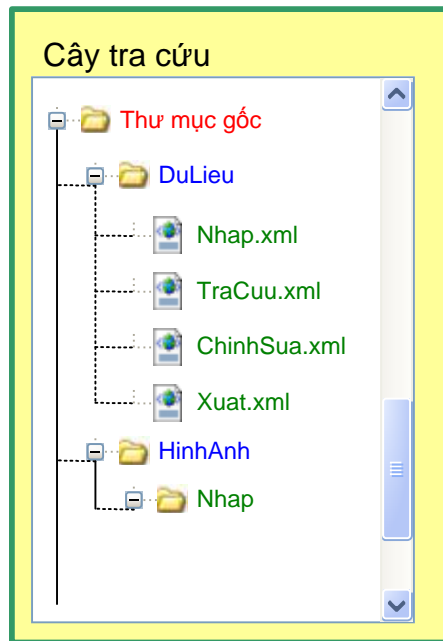
Hình 2-13 Bộ lọc cho phép kết hợp tất cả hay một vài tiêu chí – Dạng 1



The form has a yellow background and a green border. It contains three labels: 'Nhãn 1:', 'Nhãn 2:', and 'Nhãn 3:'. The 'Nhãn 1:' label is followed by a text input field with the placeholder text 'Nhập dữ liệu vào đây'. The 'Nhãn 2:' label is followed by a dropdown menu with the placeholder text 'Dữ liệu'. The 'Nhãn 3:' label is followed by two radio buttons, each followed by a label: 'Chọn lựa 1' and 'Chọn lựa 2'. Both radio buttons are selected. At the bottom of the form is a button labeled 'Lọc'.

Hình 2-14 Bộ lọc cho phép kết hợp tất cả hay một vài tiêu chí – Dạng 2

Nhóm bộ lọc 5: Sử dụng cây tra cứu để thể hiện trực quan các cấu trúc phân cấp dạng cây. Đây là một trong những bộ tra cứu phổ biến trong các giao diện dưới dạng web.



Hình 2-15 Bộ lọc dùng cây tra cứu

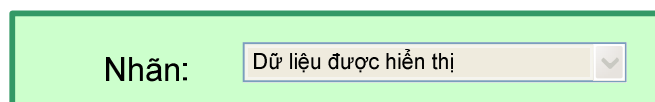
2.2.1.2 Thành phần Bộ hiển thị:

Bộ hiển thị đảm nhận chức năng hiển thị các thông tin của đối tượng trên giao diện. Các thông tin được hiển thị có thể là kết quả của một quá trình tra cứu hay những thông tin được cung cấp để hỗ trợ thêm trong quá trình tra cứu, hoặc chỉ đơn giản là những thông tin danh mục mà người sử dụng cần biết.

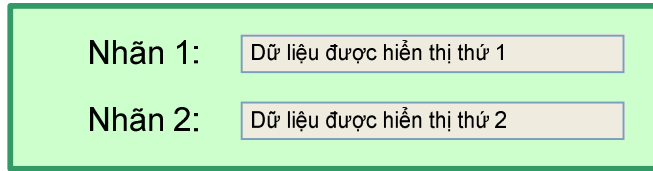
Bộ hiển thị không cho phép chỉnh sửa, thực hiện một hành động nào đó là biến đổi dữ liệu được cung cấp.

Một số loại Bộ hiển thị được khảo sát bao gồm:

Nhóm bộ hiển thị 1: dùng để hiển thị thông tin của một đối tượng.



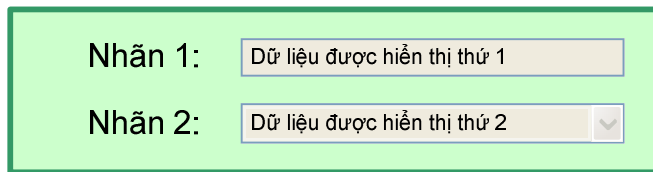
Hình 2-16 Bộ hiển thị cho một thông tin



Nhãn 1:

Nhãn 2:

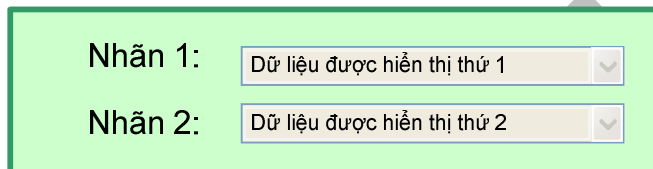
Hình 2-17 Bộ hiển thị cho nhiều thông tin – Dạng 1



Nhãn 1:

Nhãn 2:

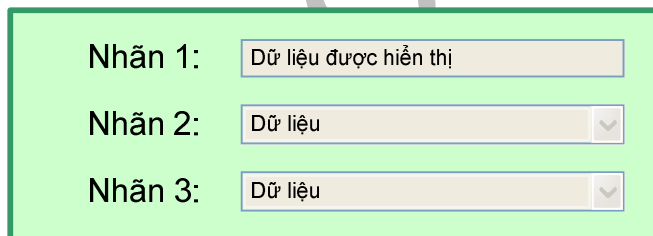
Hình 2-18 Bộ hiển thị cho nhiều thông tin – Dạng 2



Nhãn 1:

Nhãn 2:

Hình 2-19 Bộ hiển thị cho nhiều thông tin – Dạng 3

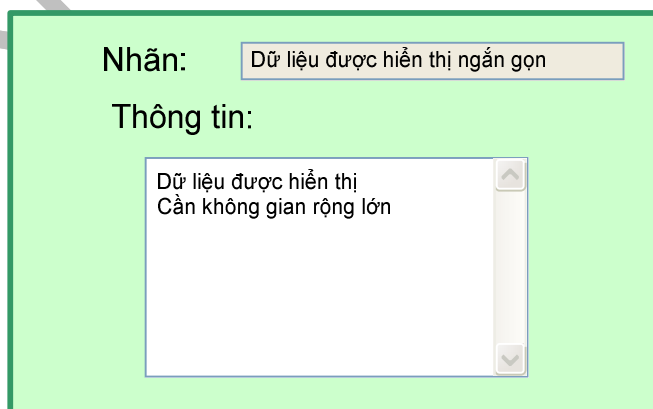


Nhãn 1:

Nhãn 2:

Nhãn 3:

Hình 2-20 Bộ hiển thị cho nhiều thông tin – Dạng 4



Nhãn:

Thông tin:

Hình 2-21 Bộ hiển thị cho nhiều thông tin – Dạng 5

Nhóm bộ hiển thị 2: để hiển thị nhiều thông tin của nhiều đối tượng trên giao diện, Bộ hiển thị này dùng lưới dữ liệu để tiết kiệm không gian, trình bày với bố cục chặt chẽ.

Thông tin:

Cột 1	Cột 2	Cột 3

Hình 2-22 Bộ hiển thị thông tin nhiều đối tượng cùng loại

Nhóm bộ hiển thị 3: là sự kết hợp của bộ hiển thị 1 và 2. Bộ hiển thị 2 được dùng như phần cung cấp các thông tin chính của tất cả các đối tượng, trong khi đó, bộ hiển thị 1 sẽ thực hiện chức năng cung cấp đầy đủ thông tin của 1 đối tượng được chọn. Dưới đây chỉ trình bày một số sự kết hợp của 2 bộ thị trên để minh họa.

Thông tin chung

Cột 1	Cột 2	Cột 3

Thông tin chi tiết

Nhãn 1:

Nhãn 2:

Nhãn 3: ☒ Chọn lựa 1 ☐ Chọn lựa 2

Hình 2-23 Bộ hiển thị kết hợp nhóm 1 và nhóm 2 – Dạng 1

Hình 2-24 Bộ hiển thị kết hợp nhóm 1 và nhóm 2 – Dạng 1

2.2.1.3 Thành phần Bộ chỉnh sửa:

Bộ chỉnh sửa đảm nhận chức năng hiển thị các thông tin của đối tượng trên giao diện và cho phép người dùng thay đổi các thông tin này.

Bộ chỉnh sửa thường được sử dụng trong các giao diện nhập (dữ liệu hiển thị ban đầu được khởi động theo ngữ cảnh), chỉnh sửa. Trong màn hình chỉnh sửa, đôi khi cũng tích hợp cả tính năng tra cứu. Khi đó, các thông tin kết quả tra cứu không chỉ để tham khảo mà còn là đầu vào cho việc chỉnh sửa.

Một số loại Bộ chỉnh sửa được khảo sát bao gồm:

Nhóm bộ chỉnh sửa 1: dùng cho việc chỉnh sửa thông tin của một đối tượng.

Hình 2-25 Bộ chỉnh sửa với 1 thông tin cần được nhập

Hình 2-26 Bộ chỉnh sửa với 1 thông tin cần được chọn – Dạng 1

Thông tin:

Dữ liệu

This form contains a label 'Thông tin:' followed by a dropdown menu. The dropdown menu is currently displaying 'Dữ liệu' and has a small blue arrow icon on the right side.

Hình 2-27 Bộ chỉnh sửa với 1 thông tin cần được chọn – Dạng 2

Nhấn 1:

Nhấn 2:

This form contains two labels, 'Nhấn 1:' and 'Nhấn 2:', each followed by a text input field. Both input fields contain the placeholder text 'Nhập dữ liệu vào đây'.

Hình 2-28 Bộ chỉnh sửa với nhiều thông tin cần được nhập

Nhấn 1:

Nhấn 2:

This form contains two labels, 'Nhấn 1:' and 'Nhấn 2:', each followed by a dropdown menu. Both dropdown menus are currently displaying 'Dữ liệu' and have a small blue arrow icon on the right side.

Hình 2-29 Bộ chỉnh sửa với nhiều thông tin cần được chọn

Nhấn 1:

Nhấn 2:

This form contains two labels, 'Nhấn 1:' and 'Nhấn 2:', each followed by a form element. 'Nhấn 1:' is followed by a text input field containing 'Nhập dữ liệu vào đây'. 'Nhấn 2:' is followed by a dropdown menu displaying 'Dữ liệu' with a small blue arrow icon on the right side.

Hình 2-30 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 1

Nhấn 1:

Nhấn 2:

Nhấn 3:

This form contains three labels, 'Nhấn 1:', 'Nhấn 2:', and 'Nhấn 3:', each followed by a form element. 'Nhấn 1:' is followed by a text input field containing 'Nhập dữ liệu vào đây'. 'Nhấn 2:' and 'Nhấn 3:' are each followed by a dropdown menu displaying 'Dữ liệu' with a small blue arrow icon on the right side.

Hình 2-31 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 2

Nhấn:

Thông tin:

Hình 2-32 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 3

Nhấn 1:

Nhấn 2:

☒ Chọn lựa 1

☒ Chọn lựa 2

☒ Chọn lựa 3

Hình 2-33 Bộ chỉnh sửa với nhiều thông tin cần được nhập và được chọn – Dạng 4

Nhóm bộ chỉnh sửa 2: được sử dụng để chỉnh sửa nhiều thông tin của nhiều đối tượng trên giao diện, Bộ chỉnh sửa này dùng lưới dữ liệu để tiết kiệm không gian, trình bày với bố cục chặt chẽ. Ngoài ra, bộ chỉnh sửa còn cho phép người dùng thao tác thêm, cập nhật, xóa lên dữ liệu tùy theo mức độ.

Thông tin

Cột 1	Cột 2	Cột 3

Hình 2-34 Bộ chỉnh sửa thông tin nhiều đối tượng cùng loại

Nhóm bộ chỉnh sửa 3: là sự kết hợp của bộ chỉnh sửa 1 và bộ hiển thị 2. Bộ chỉnh sửa 2 được dùng như phần cung cấp các thông tin chính của tất cả các đối

tượng, trong khi đó, Bộ chỉnh sửa 1 sẽ thực hiện chức năng cung cấp đầy đủ thông tin của 1 đối tượng được chọn và cho phép các thao tác cập nhật, xóa trên đối tượng này. Dưới đây chỉ trình bày một số bộ chỉnh sửa thuộc nhóm 3.

Hình 2-35 Bộ chỉnh sửa kết hợp nhóm 1 và nhóm 2 – Dạng 1

Hình 2-36 Bộ chỉnh sửa kết hợp nhóm 1 và nhóm 2 – Dạng 2

2.2.1.4 Thành phần Bộ Các nút chức năng:

Bộ Các nút chức năng sẽ tiếp nhận các yêu cầu xử lý của giao diện.

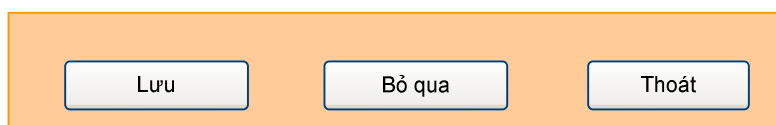
Có một số bộ nút chức năng lặp đi lặp lại trong các giao diện của hệ thống thông tin, chẳng hạn:

- Trong các giao diện nhập, ta thường bắt gặp các nút chức năng Thêm, Bỏ qua, Thoát.
- Trong giao diện chỉnh sửa, ta lại bắt gặp các nút chức năng Cập nhật, Xóa, Bỏ qua, Thoát.
- Trong giao diện tra cứu, ta lại bắt gặp các nút chức năng Lọc, Tiếp tục, Trở lại, Bỏ qua, Hoàn tất, Thoát.

Tất cả các nút chức năng trên một giao diện không nhất thiết phải tạo thành một bộ Các nút chức năng, thay vào đó, ta có thể chia chúng theo từng thành phần phù hợp với vai trò của nó trong hoạt động tổng thể. Ví dụ trong giao diện tra cứu, nút Lọc thường được gắn chung với các tiêu chí tra cứu để hình thành bộ lọc chứ không được nhóm với các nút còn lại.

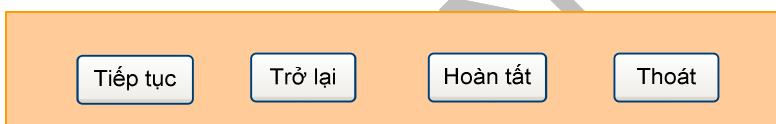
Dưới đây là một số bộ nút chức năng thường gặp khi thiết kế các giao diện trong hệ thống thông tin.

Bộ các nút chức năng 1: dùng trong giao diện nhập



Hình 2-37 Bộ các nút chức năng hỗ trợ lưu

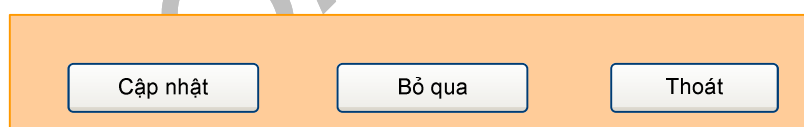
Bộ các nút chức năng 2: dùng trong giao diện tra cứu dạng wizard



Hình 2-38 Bộ các nút chức năng hỗ trợ tra cứu bằng wizard

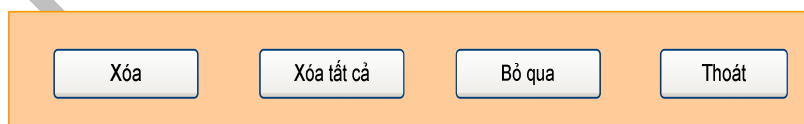
Bộ các nút chức năng 3: dùng trong giao diện chỉnh sửa

- Chỉ cho phép chỉnh sửa dữ liệu, không cho phép xóa.



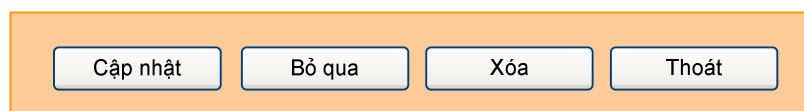
Hình 2-39 Bộ các nút chức năng hỗ trợ cập nhật

- Chỉ cho phép xóa dữ liệu, không cho phép cập nhật.



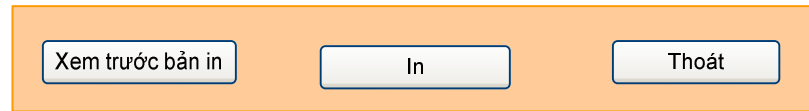
Hình 2-40 Bộ các nút chức năng hỗ trợ xóa

- Cho phép cập nhật và xóa dữ liệu.



Hình 2-41 Bộ các nút chức năng hỗ trợ cập nhật và xóa

Bộ các nút chức năng 4: dùng trong giao diện xuất.



Hình 2-42 Bộ các nút chức năng hỗ trợ kết xuất

2.2.2 Một số loại mẫu thiết kế giao diện trong hệ thống thông tin

Một số loại mẫu thiết kế giao diện thường gặp trong hệ thống thông tin đã được chúng tôi tìm hiểu là loại mẫu Simple và loại mẫu Master Detail.

Với mỗi loại mẫu Simple và Master Detail, chúng tôi đã tìm hiểu mô hình của chúng khi ứng dụng vào các giao diện nhập, tra cứu, chỉnh sửa và xuất như thế nào. Từ đó, chúng tôi tiến đến việc xây dựng mẫu thiết kế giao diện bằng cách thay thế từng phần trong mô hình bằng các thành phần đã tìm hiểu trên đây. Cuối cùng, chúng tôi gán ngữ cảnh sử dụng cho các mẫu giao diện, nói cách khác, đưa mẫu giao diện đã thiết kế vào một ngữ cảnh cụ thể, để tạo ra các thể hiện mẫu (những giao diện mà người dùng mong đợi khi sử dụng hệ thống quản lý mẫu thiết kế giao diện theo hướng thành phần).

2.2.2.1 Loại mẫu Simple

Loại mẫu Simple là mẫu thiết kế dành cho các giao diện chỉ liên quan đến một đối tượng đơn giản, thường là các danh mục trong một hệ thống thông tin. Khi ứng dụng loại mẫu Simple vào các loại giao diện nhập, tra cứu, chỉnh sửa, xuất, ta có các mô hình cụ thể.

- **Giao diện nhập dùng loại mẫu Simple:**

- o Dạng 1: các thành phần được bố trí từ trên xuống dưới

The diagram illustrates a simple input interface layout. It consists of a blue title bar at the top with the text "Tiêu đề màn hình". Below the title bar is a light blue rectangular area containing the text "TIÊU ĐỀ MÀN HÌNH". In the center is a large green rectangular area labeled "Thông tin nhập liệu". At the bottom is an orange rectangular area labeled "Các nút chức năng hỗ trợ lưu". A red arrow points from the bottom right corner of the green area to the orange area.

Hình 2-43 Simple – Nhập – Dạng 1

- o Mẫu thiết kế giao diện:

Dựa vào mô hình trên, ta sẽ lắp ghép các thành phần đã tìm hiểu để tạo ra một mẫu thiết kế giao diện dùng cho giao diện nhập.

The diagram shows a detailed design of the Simple – Nhập – Dạng 1 interface. It features a blue title bar with "Tiêu đề màn hình". Below it is a light blue header area with "TIÊU ĐỀ MÀN HÌNH". The main content area is green and contains three input fields labeled "Nhấn 1", "Nhấn 2", and "Nhấn 3". "Nhấn 1" has a text input field with placeholder "Dữ liệu nhập". "Nhấn 2" has a dropdown menu with placeholder "Dữ liệu chọn". "Nhấn 3" has a text input field with placeholder "Dữ liệu nhập". At the bottom is an orange bar containing three buttons: "Xem bản in", "In", and "Thoát".

Hình 2-44 Mẫu thiết kế giao diện của Simple – Nhập – Dạng 1

- Thể hiện mẫu:

Dựa vào mẫu thiết kế trên, ta sẽ tạo ra một thể hiện mẫu, đó là màn hình nhập khách hàng với các thông tin Tên khách hàng, Loại khách hàng và Địa chỉ.

Tiêu đề màn hình

NHẬP KHÁCH HÀNG

Tên khách hàng: Nguyễn Văn Anh

Loại khách hàng: Khách hàng thân thiết

Địa chỉ: 85/12 Nguyễn Kiệm Phú Nhuận TP.HCM

Lưu Bỏ qua Thoát

Hình 2-45 Thể hiện mẫu Nhập khách hàng – Dạng 1

- Dạng 2: các thành phần được bố trí từ trái qua phải

Tiêu đề màn hình

TIÊU ĐỀ MÀN HÌNH

Thông tin nhập liệu

Các nút chức năng hỗ trợ lưu

Hình 2-46 Simple – Nhập – Dạng 2

- Mẫu thiết kế giao diện:

The screenshot shows a web form with a blue header bar containing the text 'Tiêu đề màn hình'. Below the header is a light blue box with the title 'TIÊU ĐỀ MÀN HÌNH'. The main content area is divided into two sections. The left section, with a light green background, contains three labels: 'Nhấn 1', 'Nhấn 2', and 'Nhấn 3'. Each label is followed by an input field. The first and third fields are text boxes with the placeholder 'Dữ liệu nhập'. The second field is a dropdown menu with the placeholder 'Dữ liệu chọn'. The right section, with a light orange background, contains three buttons: 'Lưu', 'Bỏ qua', and 'Thoát'.

Hình 2-47 Mẫu thiết kế giao diện của Simple – Nhập – Dạng 2

- Thể hiện mẫu:

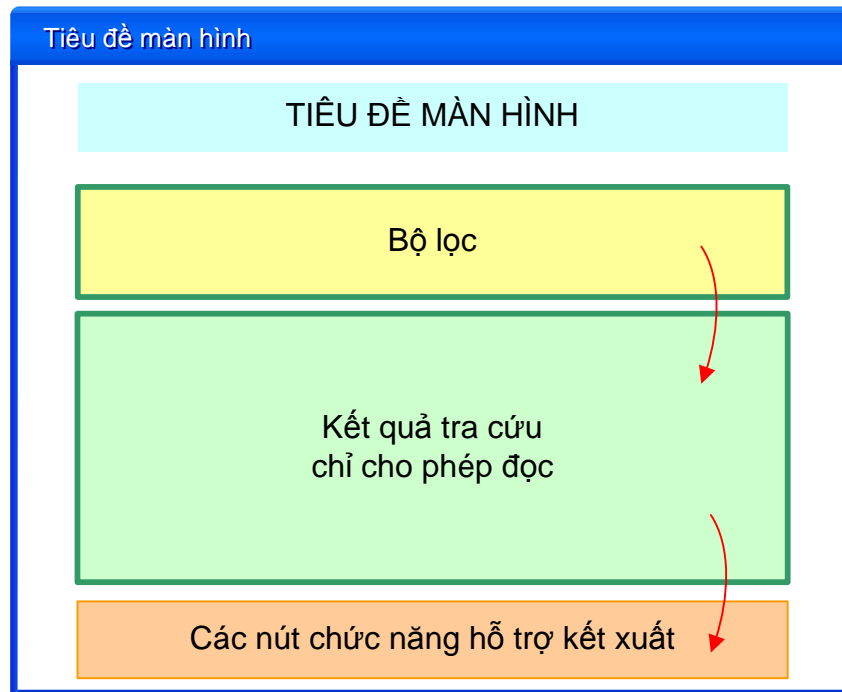
The screenshot shows a web form with a blue header bar containing the text 'Tiêu đề màn hình'. Below the header is a light blue box with the title 'NHẬP KHÁCH HÀNG'. The main content area is divided into two sections. The left section, with a light green background, contains three labels: 'Tên khách hàng', 'Loại khách hàng', and 'Địa chỉ'. Each label is followed by an input field. The first field is a text box with the value 'Nguyễn Văn Anh'. The second field is a dropdown menu with the value 'Khách hàng thân thiết'. The third field is a text box with the value '88/1 Nguyễn Kiệm Phú Nhuận'. The right section, with a light orange background, contains three buttons: 'Lưu', 'Bỏ qua', and 'Thoát'.

Hình 2-48 Thể hiện mẫu Nhập khách hàng – Dạng 2

Tương tự như giao diện Nhập, các giao diện khác cũng có nhiều dạng khác nhau. Các dạng này có cùng nội dung nhưng khác cách bố trí. Để vấn đề trình bày được súc tích, ở những phần sau chúng tôi chỉ chọn ra một dạng thông dụng nhất, đi kèm là một mẫu thiết kế giao diện và một thể hiện mẫu để minh họa.

- **Giao diện Tra cứu dùng mẫu Simple:**

- Mô hình:



Hình 2-49 Simple – Tra cứu

- Mẫu thiết kế giao diện:

Tiêu đề màn hình

TIÊU ĐỀ MÀN HÌNH

Nhân

Cột 1	Cột 2	Cột 3	Cột 4

Hình 2-50 Mẫu thiết kế giao diện của Simple – Tra cứu

- Thể hiện mẫu:

Tiêu đề màn hình

TRA CỨU KHÁCH HÀNG

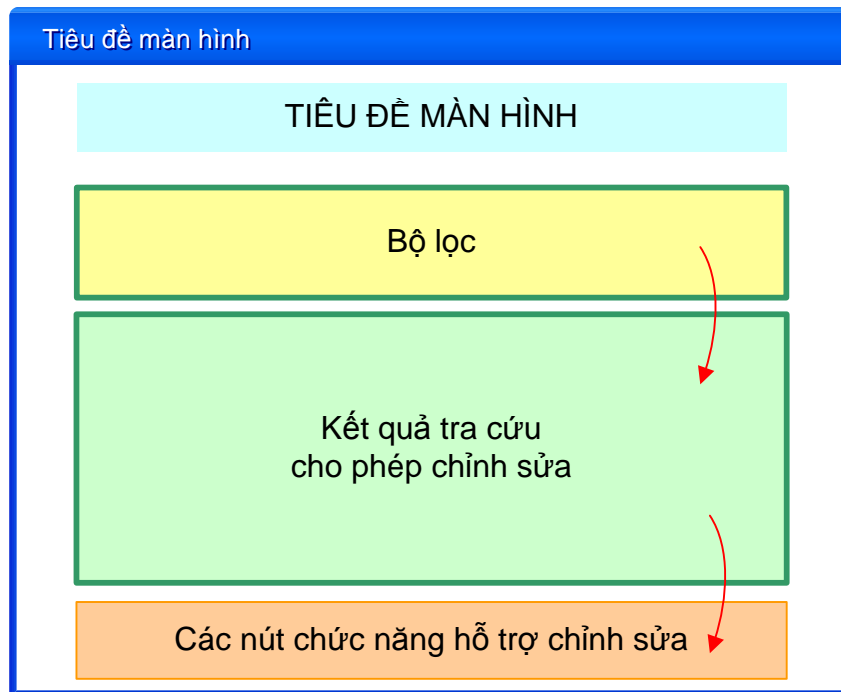
Tên

STT	Tên	Loại khách hàng	Địa chỉ
1	Nguyễn Văn Anh	Khách hàng thân	88/1 N
2	Trần Nguyễn Phương La		
3	Nguyễn Thị Hoa		
4	Nguyễn Thị Thu		

Hình 2-51 Thể hiện mẫu Tra cứu khách hàng

- **Giao diện chỉnh sửa dùng mẫu Simple:**

- o Mô hình:



Hình 2-52 Simple – Chỉnh sửa

- o Mẫu thiết kế giao diện:

The detailed design of the Simple editing interface is shown. It features a blue title bar "Tiêu đề màn hình" and a light blue header "TIÊU ĐỀ MÀN HÌNH". Below the header is a yellow box labeled "Nhấn" containing a text input field "Dữ liệu nhập" and a "Lọc" button. Underneath is a green box containing a table with 4 columns labeled "Cột 1", "Cột 2", "Cột 3", and "Cột 4". The table has 10 rows. To the right of the table are vertical scroll arrows. Below the table is a horizontal scroll bar. At the bottom is an orange box containing four buttons: "Cập nhật", "Bỏ qua", "Xóa", and "Thoát".

Hình 2-53 Mẫu thiết kế giao diện của Simple – Chỉnh sửa

- Thể hiện mẫu:

Tiêu đề màn hình

CHỈNH SỬA THÔNG TIN KHÁCH HÀNG

Tên

STT	Tên	Loại khách hàng	Địa chỉ
1	Nguyễn Văn Anh	Khách hàng thân	88/1 N
2	Trần Nguyễn Phương La		
3	Nguyễn Thị Hoa		
4	Nguyễn Thị Thu		

Hình 2-54 Thể hiện mẫu Chỉnh sửa thông tin khách hàng

- Giao diện xuất dùng mẫu Simple:

- Mô hình:

Tiêu đề màn hình

TIÊU ĐỀ MÀN HÌNH

Thông tin hiển thị
chỉ cho phép đọc

Các nút chức năng hỗ trợ kết xuất

Hình 2-55 Simple – Xuất

- Mẫu thiết kế giao diện:

The screenshot shows a web form with a blue header bar containing the text 'Tiêu đề màn hình'. Below the header is a light blue box with the title 'TIÊU ĐỀ MÀN HÌNH'. The main content area is a green box containing three labels: 'Nhãn 1', 'Nhãn 2', and 'Nhãn 3'. Each label is followed by a text input field. The first two fields contain the placeholder text 'Dữ liệu hiển thị', and the second field has a dropdown arrow. Below the green box is an orange bar containing three buttons: 'Xem bản in', 'In', and 'Thoát'.

Hình 2-56 Mẫu thiết kế giao diện của Simple – Xuất

- Thể hiện mẫu:

The screenshot shows a web form with a blue header bar containing the text 'Tiêu đề màn hình'. Below the header is a light blue box with the title 'NHẬP KHÁCH HÀNG'. The main content area is a green box containing three labels: 'Tên khách hàng', 'Loại khách hàng', and 'Địa chỉ'. Each label is followed by a text input field. The first field contains the sample text 'Nguyễn Văn Anh', the second field contains 'Khách hàng thân thiết' with a dropdown arrow, and the third field contains '85/12 Nguyễn Kiệm Phú Nhuận TP.HCM'. Below the green box is an orange bar containing three buttons: 'Xem bản in', 'In', and 'Thoát'.

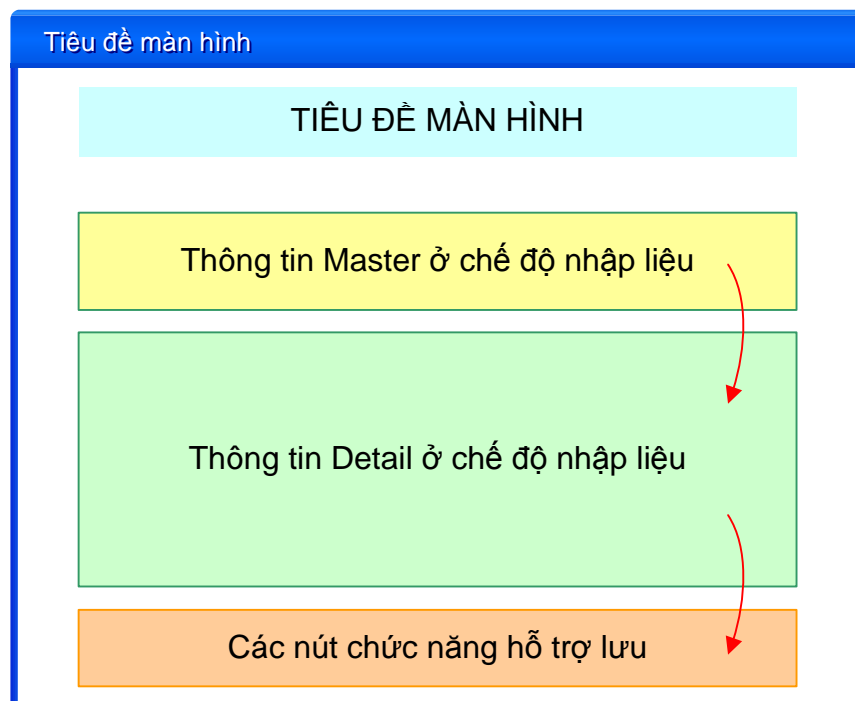
Hình 2-57 Thể hiện mẫu Xuất thông tin khách hàng

2.2.2.2 Loại mẫu Mater Detail:

Cùng với loại mẫu Simple, loại mẫu Master Detail cũng được sử dụng phổ biến trong các hệ thống thông tin. Loại mẫu Master Detail là lựa chọn lý tưởng khi thiết kế giao diện để quản lý các đối tượng có mối quan hệ chính phụ, cha con, chẳng hạn đối tượng Hóa đơn và Chi tiết hóa đơn.

- **Giao diện Nhập:**

- o Mô hình:



Hình 2-58 Master Detail – Nhập

- Mẫu thiết kế giao diện:

Tiêu đề màn hình

TIÊU ĐỀ MÀN HÌNH

Nhãn

Cột 1	Cột 2	Cột 3	Cột 4

Lưu Bỏ qua Thoát

Hình 2-59 Mẫu thiết kế giao diện của Master Detail – Nhập

- Thể hiện mẫu:

Tiêu đề màn hình

NHẬP DANH SÁCH NHÂN VIÊN

Phòng ban

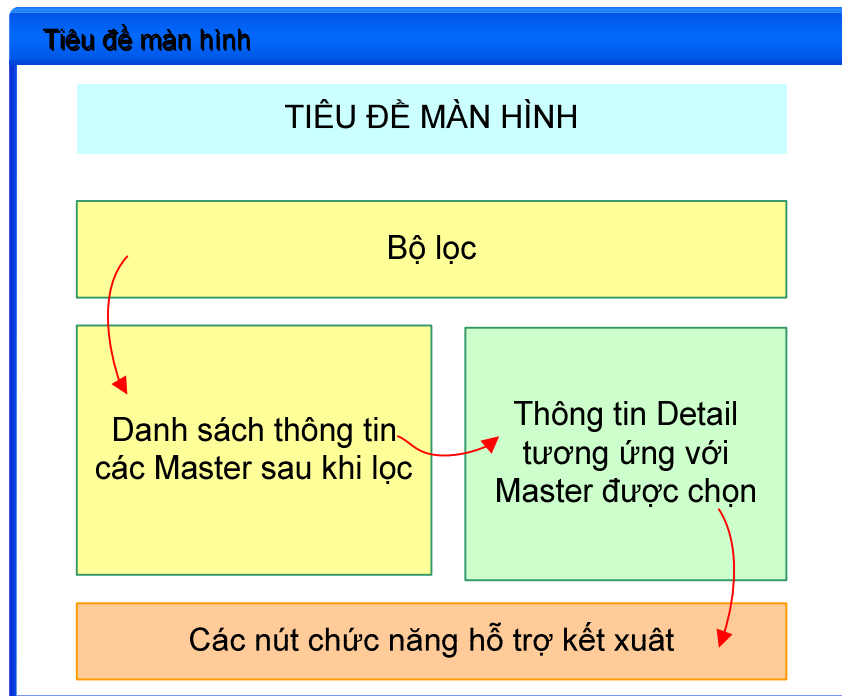
STT	Họ tên	Ngày sinh	Chức vụ
1	Nguyễn Văn Anh	1/1/1970	Trưởng p
2	Lê Thị Hồng	2/2/1980	Phó phó
3	Trần Chí Trung	3/3/1983	Nhân viên

Lưu Bỏ qua Thoát

Hình 2-60 Thể hiện mẫu Nhập Danh sách nhân viên

- **Giao diện Tra cứu:**

- o Mô hình:



Hình 2-61 Master Detail – Tra cứu

- o Mẫu thiết kế giao diện:

The screenshot shows the design of the Master Detail search interface. It has a blue header bar with "Tiêu đề màn hình". Below it is a light blue box with "TIÊU ĐỀ MÀN HÌNH". A yellow box labeled "Nhãn" (Label) contains a dropdown menu with "Dữ liệu" (Data). Below this is a table with two columns, "Cột 1" and "Cột 2", and several rows. To the right of the table is a green box containing three labels: "Nhãn 1", "Nhãn 2", and "Nhãn 3", each followed by a dropdown menu with "Dữ liệu hiển thị" (Display data). At the bottom, an orange box contains three buttons: "Xem bản in" (Print), "In" (Print), and "Thoát" (Exit).

Hình 2-62 Mẫu thiết kế giao diện của Master Detail – Tra cứu

- Thể hiện mẫu:

Cột 1	Cột 2
1	Nguyễn Văn Anh
2	Lê Thị Hồng
3	Trần Chí Trung

Hình 2-63 Thể hiện mẫu Tra cứu Nhân viên

- **Giao diện Chỉnh sửa:**

- Mô hình:



Hình 2-64 Master Detail – Chỉnh sửa

- Mẫu thiết kế giao diện:

Tiêu đề màn hình

TIÊU ĐỀ MÀN HÌNH

Nhãn

Cột 1	Cột 2

Nhãn 1
Nhãn 2
Nhãn 3

Hình 2-65 Mẫu thiết kế giao diện của Master Detail – Chỉnh sửa

- Thể hiện mẫu:

Tiêu đề màn hình

CHỈNH SỬA THÔNG TIN NHÂN VIÊN

Phòng ban

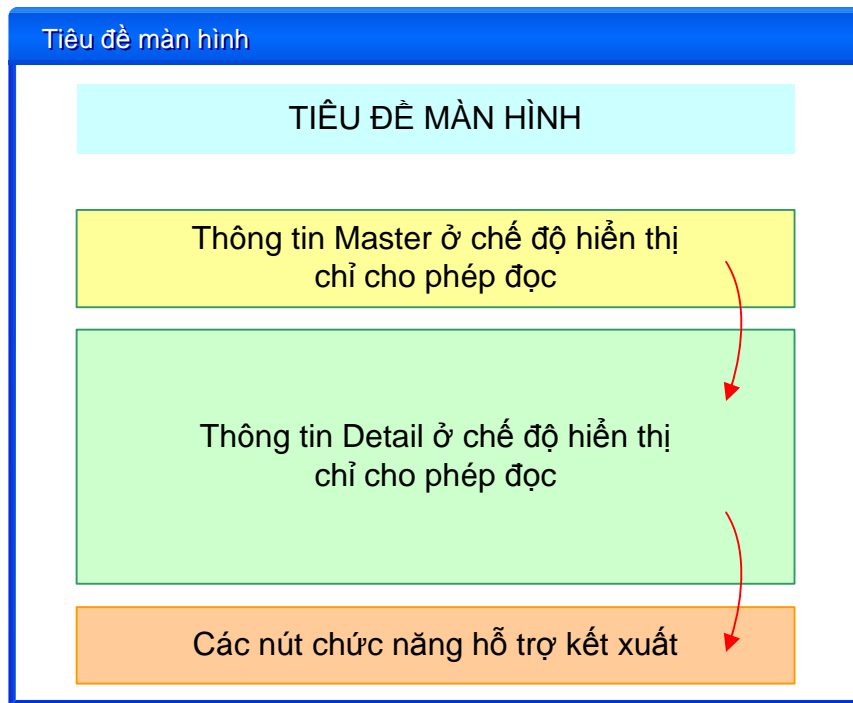
Cột 1	Cột 2
1	Nguyễn Văn Anh
2	Lê Thị Hồng
3	Trần Chí Trung

Tên
Ngày sinh
Chức vụ

Hình 2-66 Thể hiện mẫu Chỉnh sửa thông tin nhân viên

- **Giao diện Xuất:**

o Mô hình:



Hình 2-67 Master Detail – Xuất

o Mẫu thiết kế giao diện:

The screenshot shows the actual design of the Master Detail - Export interface. It features a blue title bar with the text "Tiêu đề màn hình". Below the title bar is a light blue header section labeled "TIÊU ĐỀ MÀN HÌNH". The main content area is divided into two sections: a yellow section labeled "Nhãn" with a dropdown menu labeled "Dữ liệu", and a green section containing a table with four columns labeled "Cột 1", "Cột 2", "Cột 3", and "Cột 4". The table has multiple rows and includes scrollbars. At the bottom of the interface is an orange section containing three buttons: "Xem bản in", "In", and "Thoát".

Hình 2-68 Mẫu thiết kế giao diện của Master Detail – Xuất

- Thể hiện mẫu:

Tiêu đề màn hình

XUẤT DANH SÁCH NHÂN VIÊN

Phòng ban

STT	Họ tên	Ngày sinh	Chức vụ
1	Nguyễn Văn Anh	1/1/1970	Trưởng p
2	Lê Thị Hồng	2/2/1980	Phó phó
3	Trần Chí Trung	3/3/1983	Nhân viên

Xem bản in In Thoát

Hình 2-69 Thể hiện mẫu Xuất danh sách nhân viên

2.3 Các mẫu thiết kế

Trong quá trình thực hiện luận văn, chúng tôi đã gặp phải 2 vấn đề:

- Tổ chức các điều khiển để hỗ trợ thiết kế giao diện bằng đồ họa.
- Hỗ trợ khả năng plug-in:
 - o Các module nhập vào giao diện từ một ứng dụng viết trên 1 ngôn ngữ lập trình cụ thể vào hệ thống,
 - o Các module xuất ra mẫu và thể hiện mẫu thành giao diện cho một ứng dụng viết trên 1 ngôn ngữ lập trình cụ thể.

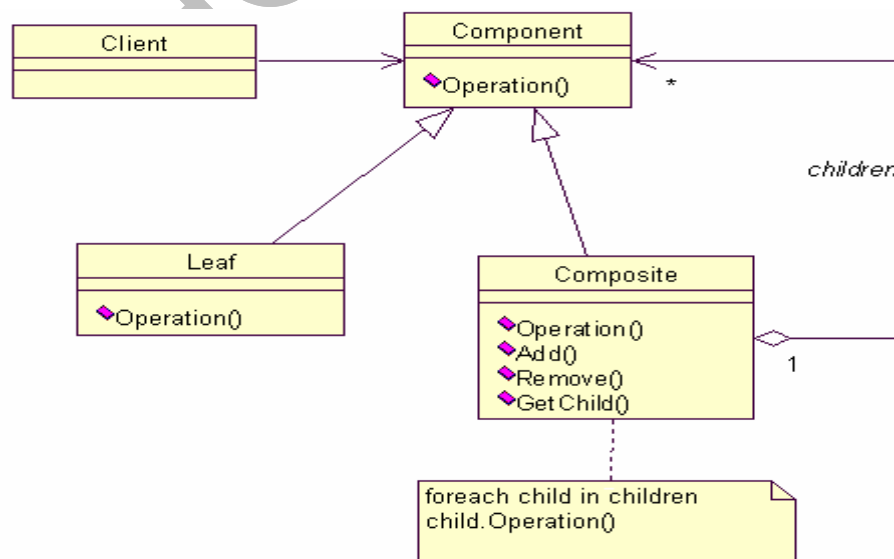
Để giải quyết, chúng tôi đã tham khảo 23 mẫu thiết kế GoF và ứng dụng được 2 mẫu thiết kế Composite và Proxy. Sau đây là phần trình bày về 2 mẫu thiết kế được sử dụng.

2.3.1 Mẫu Composite

2.3.1.1 Lý thuyết

Composite tạm dịch là “đối tượng đa hợp”, thuộc lớp mẫu cấu trúc đối tượng. Ý tưởng: tổ chức các đối tượng theo cấu trúc phân cấp dạng cây. Tất các đối tượng trong cấu trúc được thao tác một cách thuần nhất như nhau.

Sơ đồ UML:



Hình 2-70 Sơ đồ UML cho mẫu Composite

Các thành viên

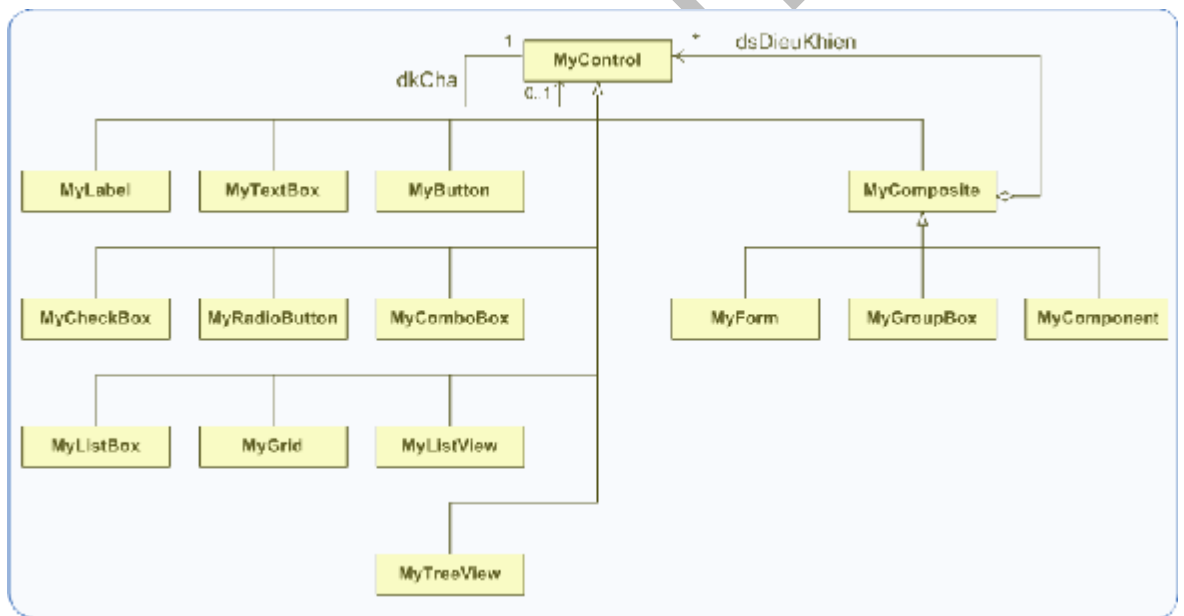
- **Component:**
 - Khai báo một giao diện chung cho các đối tượng .
 - Thực thi hành vi mặc định của giao diện chung cho tất cả các lớp khi thích hợp.
 - Khai báo một giao diện để truy cập và quản lý các thành phần con của nó.
 - (Tùy chọn) Định nghĩa một giao diện để truy cập đến cha của một thành phần trong cấu trúc đệ quy và thực thi nó nếu phù hợp.
- **Leaf:**
 - Thể hiện các đối tượng lá trong cấu trúc cây. Một đối tượng lá là một đối tượng không có con.
 - Định nghĩa hành vi cho các đối tượng gốc trong cấu trúc cây.
- **Composite:**
 - Định nghĩa hành vi cho các thành phần có đối tượng con.
 - Lưu trữ các thành phần con
 - Thực thi những xử lý liên quan đến thành phần con trong giao diện của lớp đối tượng Component
- **Client:** xử lý các đối tượng trong cấu trúc cây thông qua giao diện của lớp đối tượng Component.

2.3.1.2 Áp dụng vào hệ thống quản lý mẫu

Để đáp ứng yêu cầu hỗ trợ thiết kế giao diện bằng đồ họa, chúng tôi đã tiến hành tìm hiểu cách thức tổ chức của các điều khiển trong các ngôn ngữ lập trình C# và VB 6.0. Chúng tôi đã có các nhận xét rút ra như sau:

- Cần hỗ trợ 11 điều khiển chuẩn để thiết kế giao diện:

- Label
- Button
- TextBox
- CheckBox
- RadioButton
- GroupBox
- ListBox
- ComboBox
- Grid
- ListView
- TreeView



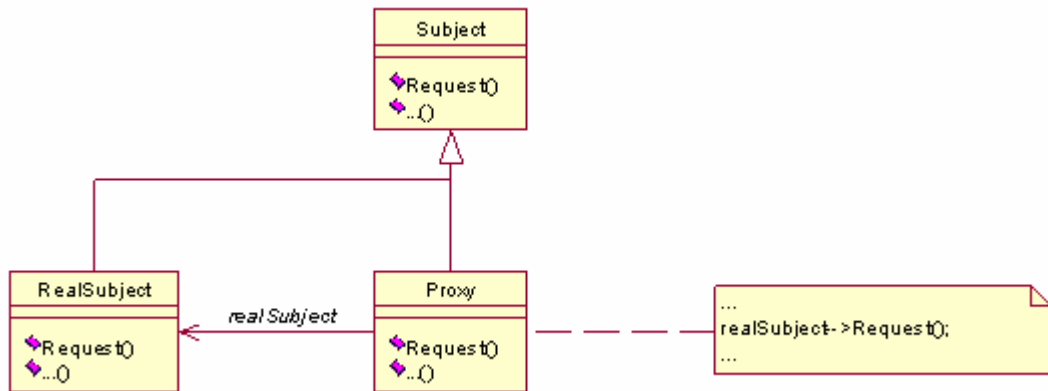
Hình 2-71 Mô hình triển khai mẫu composite trong hệ thống

2.3.2 Mẫu Proxy

2.3.2.1 Lý thuyết mẫu Proxy

Mục đích: Truy xuất một đối tượng thông qua một đối tượng được ủy nhiệm.

Sơ đồ UML:

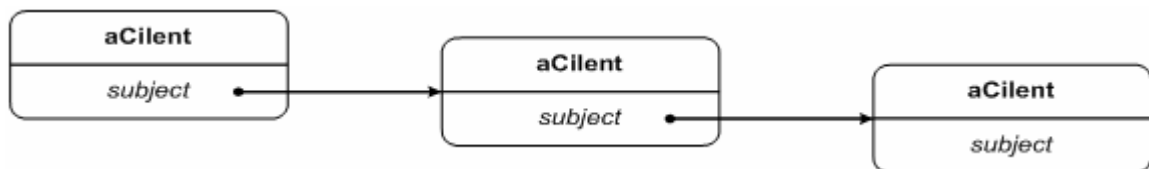


Hình 2-72 Sơ đồ UML mẫu Proxy

Các thành viên

- **Proxy:**
 - o Nắm giữ một tham chiếu cho phép proxy có thể truy xuất đến đối tượng thật muốn truy xuất
 - o Cung cấp một interface đồng nhất đến đối tượng thật sự muốn truy xuất vì thế proxy có thể xem như là đối tượng thay thế cho đối tượng thật.
 - o Kiểm soát sự truy xuất đến đối tượng thật và chịu trách nhiệm trong việc tạo ra và phá hủy nó.
- **Subject:** định nghĩa một interface cho RealSubject và Proxy từ đó proxy có thể được sử dụng ở bất kỳ đâu cần sử dụng RealSubject.
- **Real subject:** định nghĩa một đối tượng thật mà proxy làm đại diện.

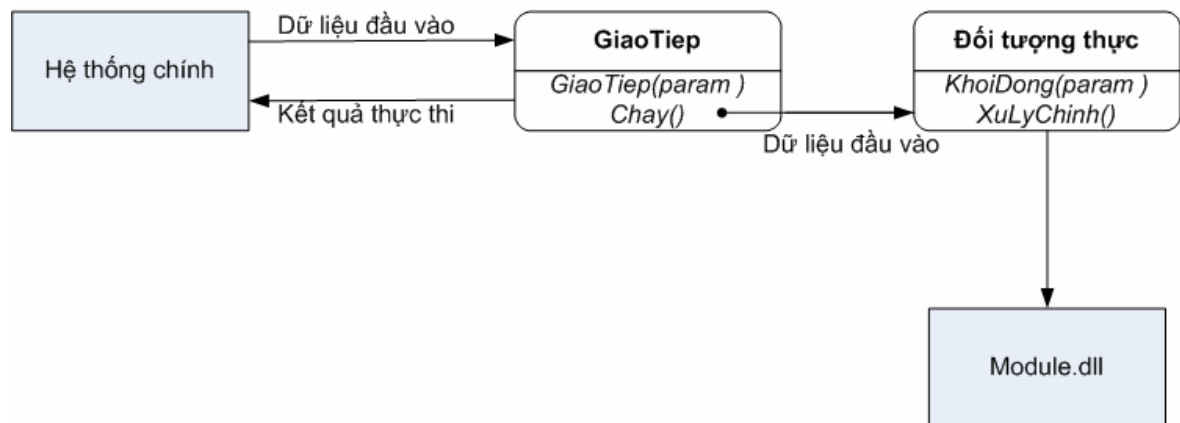
Thời điểm thực thi:



Hình 2-73 Mô hình mẫu Proxy tại thời điểm thực thi

2.3.2.2 Áp dụng vào hệ thống quản lý mẫu

Nhận thấy để thực hiện ý tưởng cho các module khác nhau được plug – in vào hệ thống. Bên trong các module, ngoài các đối tượng xử lý chính còn cần thêm một đối tượng để đảm nhận vai trò giao tiếp giữa module với hệ thống quản lý mẫu. Nhờ vậy hệ thống chính sẽ không phải quan tâm đến các đối tượng thật bên trong module, mọi giao tiếp sẽ được thực hiện thông qua đối tượng được ủy nhiệm.



Hình 2-74 Mô hình thể hiện ý tưởng thiết kế theo mẫu Proxy

Chương 3 Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng hệ thống thông tin

Nội dung

Chương này sẽ trình bày quá trình phân tích và thiết kế hệ thống quản lý mẫu mẫu thiết kế giao diện và thành phần từ mẫu thiết kế giao diện một cách rõ ràng, súc tích.

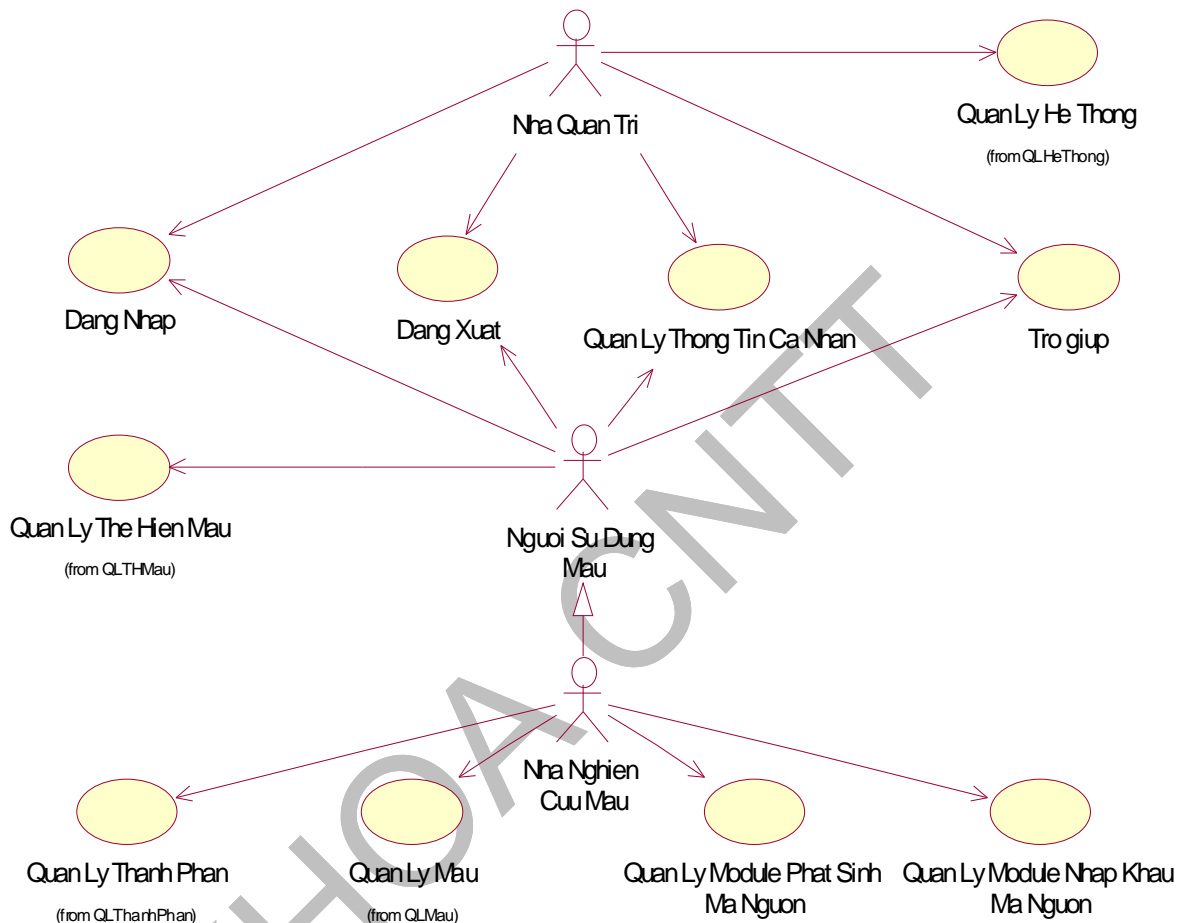
Quá trình phân tích sẽ tập trung vào các sơ đồ Use-Case và việc phân tích các chức năng quan trọng.

Quá trình thiết kế sẽ trình bày kiến trúc tổng thể, các sơ đồ tuần tự-sơ đồ cộng tác của các xử lý phức tạp, cách thiết kế lớp và kiến trúc lưu trữ được sử dụng.

3.1 Phân tích

3.1.1 Các sơ đồ Use-Case

3.1.1.1 Sơ đồ Use-Case của toàn hệ thống



Hình 3-1 Sơ đồ Use-Case của toàn hệ thống

Bảng thuyết minh Sơ đồ Use-Case của toàn hệ thống

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Quan Tri	Actor	Nhà quản trị	
2	Nha Nghien Cuc Mau	Actor	Nhà nghiên cứu mẫu	
3	Ngươi Su Dung Mau	Actor	Người sử dụng mẫu	
4	Dang Nhap	Use-Case	Đăng nhập vào hệ thống theo quyền, người dùng và mật khẩu	
5	Dang Xuat	Use-Case	Đăng xuất khỏi hệ thống	
6	Quan Ly Thong Tin Ca Nhan	Use-Case	Quản lý thông tin cá nhân của người dùng, bao gồm các chức	

			năng xem thông tin, chỉnh sửa (cập nhật, xóa).	
7	Tro Giup	Use-Case	Trợ giúp người dùng, bao gồm chức năng xem thông tin chương trình, tác giả, hướng dẫn sử dụng	
8	Quan Ly He Thong	Use-Case	Quản lý hệ thống, bao gồm các chức năng quản lý người dùng, quyền	
9	Quan Ly Thanh Phan	Use-Case	Quản lý thành phần, bao gồm các chức năng mở, lưu đặc tả, tạo mới hoàn toàn, tạo mới dựa vào một thành phần đã có, tạo biến thể, tra cứu, thêm, cập nhật, xóa thành phần.	
10	Quan Ly Mau	Use-Case	Quản lý mẫu, bao gồm các chức năng mở, lưu đặc tả, nhập khẩu mẫu từ tập tin mã nguồn, tạo mới mẫu hoàn toàn hay dựa vào một mẫu đã có, tạo biến thể, tạo thể hiện cho mẫu; tra cứu, thêm, cập nhật, xóa mẫu; thêm bớt thành phần hay lưu thành phần trong mẫu; phát sinh mã nguồn cho mẫu	
11	Quan Ly The Hien Mau	Use-Case	Quản lý thể hiện mẫu, bao gồm các chức năng mở, lưu đặc tả, nhập khẩu thể hiện mẫu từ một tập tin mã nguồn; tạo mới thể hiện mẫu hoàn toàn hay dựa vào một mẫu gốc đã có; tra cứu, thêm, xóa, cập nhật; thêm bớt thành phần, lưu thành phần trong thể hiện; phát sinh mã nguồn cho thể hiện mẫu.	
12	Quan Ly Module Phat Sinh Ma Nguon	Use-Case	Quản lý module phát sinh mã nguồn, bao gồm các chức năng thêm, cập nhật, xóa module	
13	Quan Ly Module Nhap Khau Ma Nguon	Use-Case	Quản lý module nhập khẩu tập tin mã nguồn trở thành mẫu hay thể hiện mẫu. Việc quản lý bao gồm các chức năng thêm, cập nhật, xóa module	

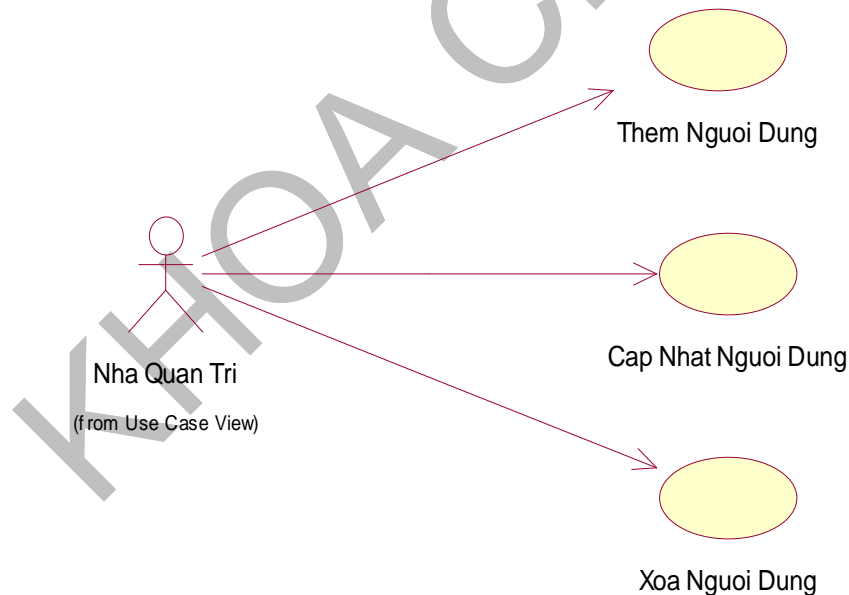
Bảng 3-1 Bảng thuyết minh Sơ đồ Use-Case của toàn hệ thống

Do các chức năng của hệ thống rất nhiều, chúng tôi đã tổ chức các chức năng thuộc cùng một nhóm với nhau. Các nhóm này bao gồm:

- Quản lý hệ thống
- Quản lý thông tin cá nhân
- Quản lý thành phần
- Quản lý mẫu
- Quản lý thể hiện mẫu
- Quản lý module nhập khẩu mã nguồn
- Quản lý module phát sinh mã nguồn.

Các chức năng khác không có độ phức tạp lớn nên vẫn được mô tả trực tiếp trong sơ đồ Use-Case của hệ thống. Sau đây là phần mô tả các Use-Case của các nhóm vừa liệt kê.

3.1.1.2 Sơ đồ Use-Case Quản lý hệ thống

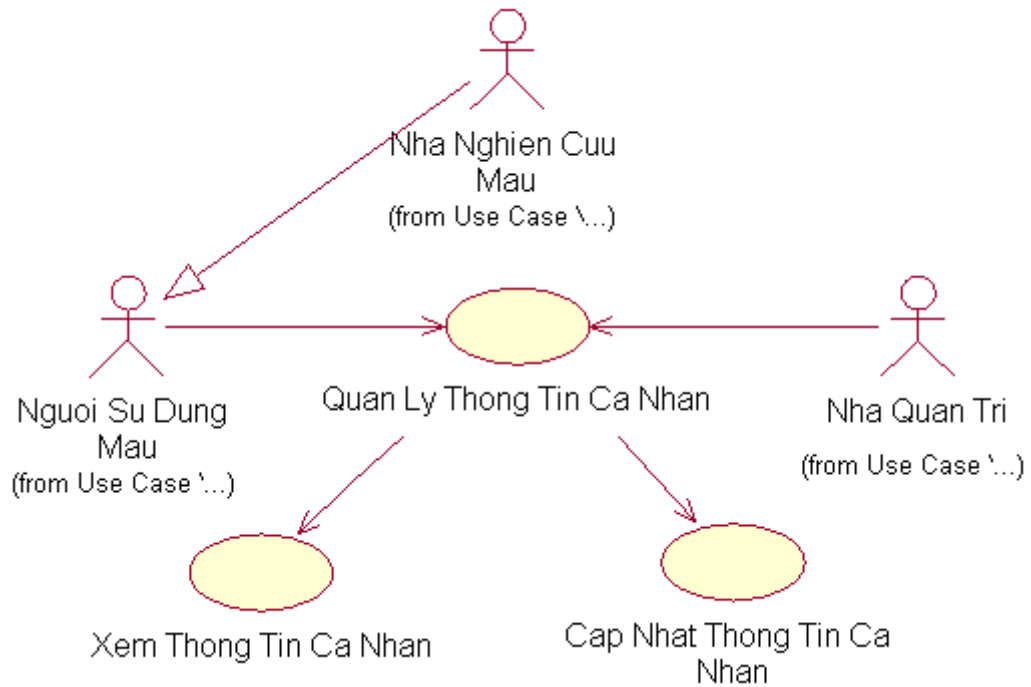


Hình 3-2 Sơ đồ Use-Case Quản lý hệ thống

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Quan Tri	Actor	Nhà quản trị	
5	Them Ngui Dung	Use-Case	Thêm người dùng	
6	Cap Nhat Ngui Dung	Use-Case	Cập nhật thông tin người dùng	
7	Xoa Ngui Dung	Use-Case	Xóa người dùng	

Bảng 3-2 Bảng thuyết minh Sơ đồ Use-Case Quản lý hệ thống

3.1.1.3 Sơ đồ Use-Case Quản lý thông tin cá nhân

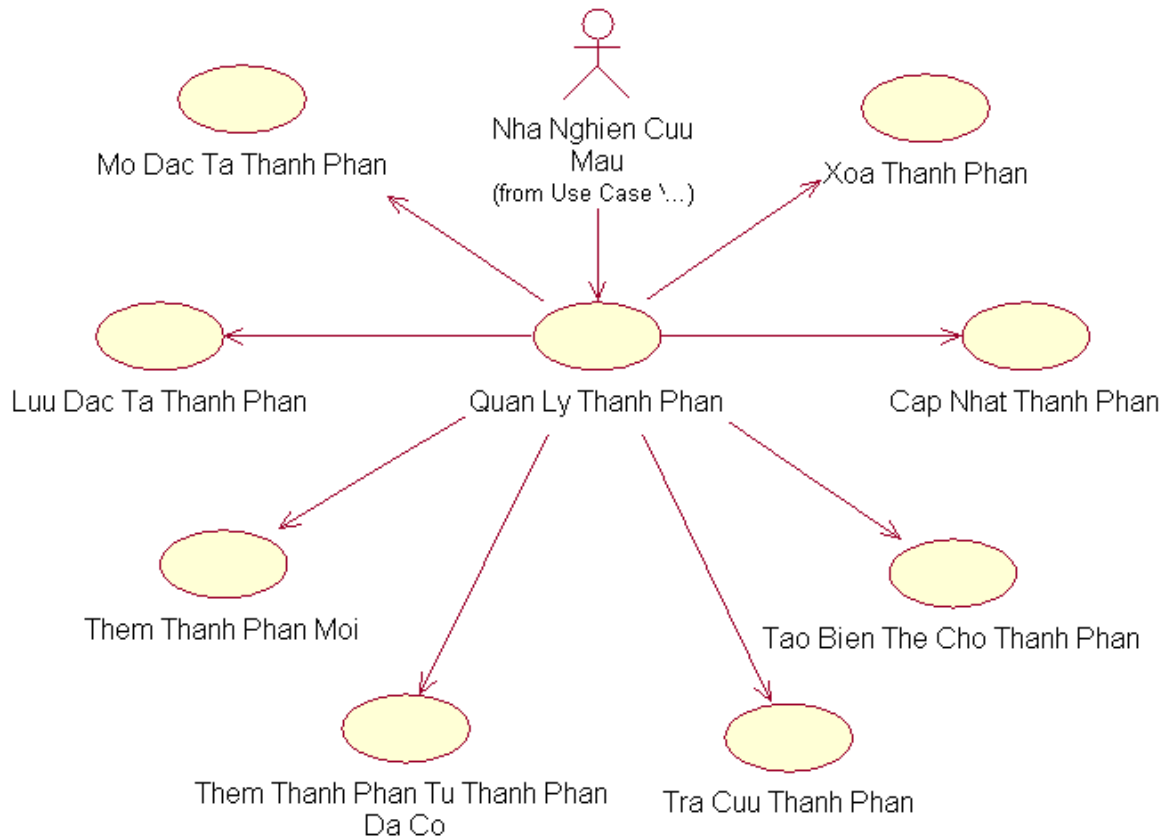


Hình 3-3 Sơ đồ Use-Case Quản lý thông tin cá nhân

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Quan Tri	Actor	Nhà quản trị	
2	Nha Nghiem Cuc Mau	Actor	Nhà nghiên cứu mẫu	
3	Nguoi Su Dung Mau	Actor	Người sử dụng mẫu	
4	Quan Ly Thong Tin Ca Nhan	Use-Case	Quản lý thông tin cá nhân	
5	Xem Thong Tin Ca Nhan	Use-Case	Xem thông tin cá nhân	
6	Cap Nhat Thong Tin Ca Nhan	Use-Case	Cập nhật thông tin cá nhân	

Bảng 3-3 Bảng thuyết minh Sơ đồ Use-Case Quản lý thông tin cá nhân

3.1.1.4 Sơ đồ Use-Case Quản lý thành phần

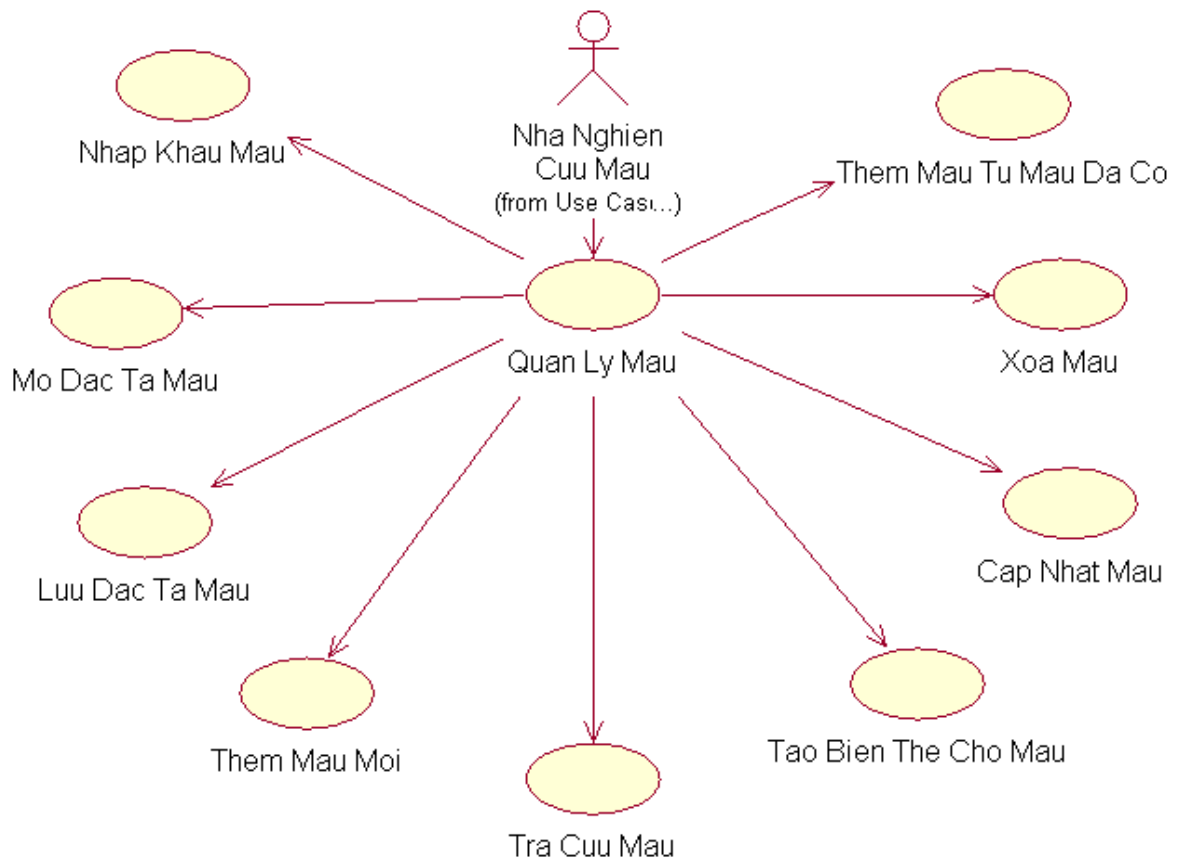


Hình 3-4 Sơ đồ Use-Case Quản lý thành phần

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Nghien Cuu Mau	Actor	Nhà nghiên cứu mẫu	
2	Quan Ly Thanh Phan	Use-Case	Quản lý thành phần	
3	Mo Dac Ta Thanh Phan	Use-Case	Mở tập tin đặc tả của thành phần	
4	Luu Dac Ta Thanh Phan	Use-Case	Lưu tập tin đặc tả của thành phần	
5	Them Thanh Phan Moi	Use-Case	Thêm thành phần mới	
6	Them Thanh Phan Tu Thanh Phan Da Co	Use-Case	Thêm thành phần từ thành phần đã có	
7	Tra Cuu Thanh Phan	Use-Case	Tra cứu thành phần	
8	Tao Bien The Cho Thanh Phan	Use-Case	Tạo biến thể cho thành phần	
9	Cap Nhat Thanh Phan	Use-Case	Cập nhật thành phần	
10	Xoa Thanh Phan	Use-Case	Xóa thành phần	

Bảng 3-4 Bảng thuyết minh Sơ đồ Use-Case Quản lý thành phần

3.1.1.5 Sơ đồ Use-Case Quản lý mẫu

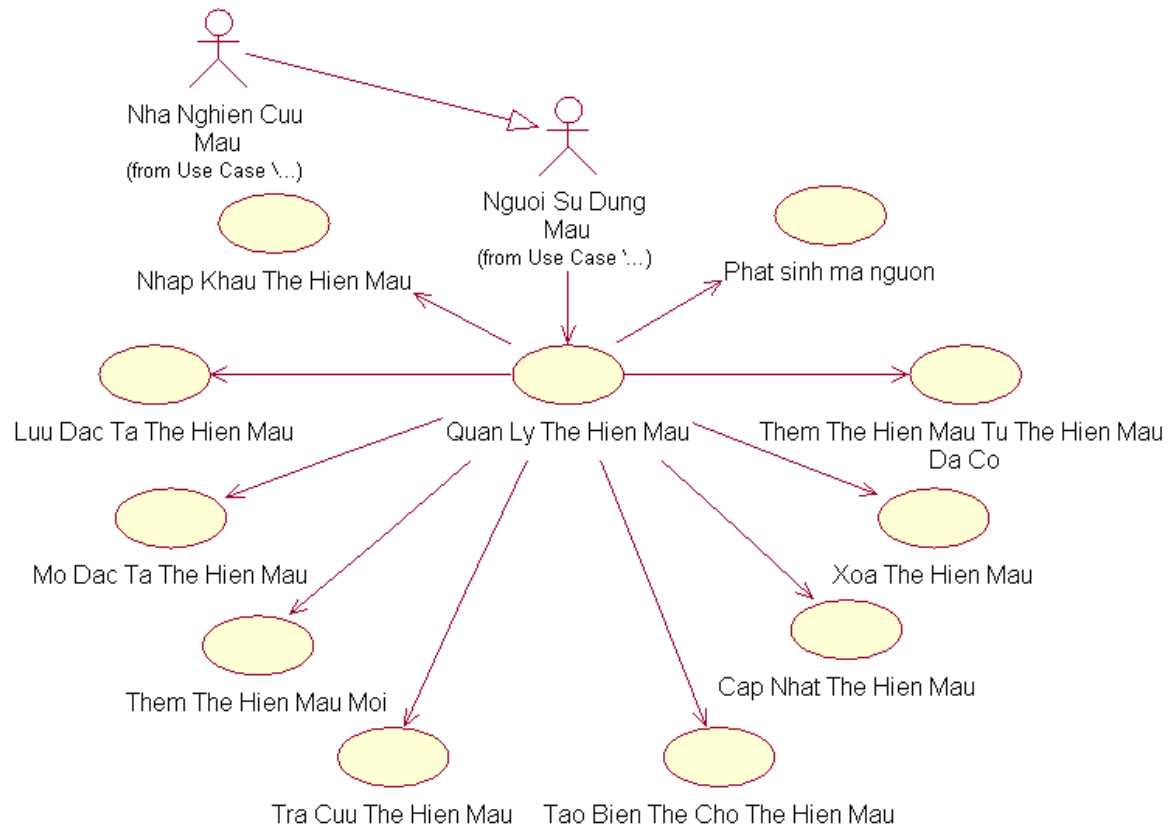


Hình 3-5 Sơ đồ Use-Case Quản lý mẫu

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nhà Nghiên Cứu Mẫu	Actor	Nhà nghiên cứu mẫu	
2	Quản Lý Mẫu	Use-Case	Quản lý mẫu	
3	Mở Tập Tin Đặc Tính Mẫu	Use-Case	Mở tập tin đặc tính của mẫu	
4	Lưu Đặc Tính Mẫu	Use-Case	Lưu đặc tính của mẫu	
5	Thêm Mẫu Mới	Use-Case	Thêm mẫu mới	
6	Thêm Mẫu Từ Mẫu Đã Có	Use-Case	Thêm mẫu từ mẫu đã có	
7	Tra Cứu Mẫu	Use-Case	Tra cứu mẫu	
8	Tạo Biên Thể Cho Mẫu	Use-Case	Tạo biên thể cho mẫu	
9	Cập Nhật Mẫu	Use-Case	Cập nhật mẫu	
10	Xóa Mẫu	Use-Case	Xóa mẫu	
11	Tạo Thể Hiện Cho Mẫu	Use-Case	Tạo thể hiện cho mẫu	
12	Phát Sinh Mã Nguồn	Use-Case	Phát sinh mã nguồn	

Bảng 3-5 Bảng thuyết minh Sơ đồ Use-Case Quản lý mẫu

3.1.1.6 Sơ đồ Use-Case Quản lý thể hiện mẫu

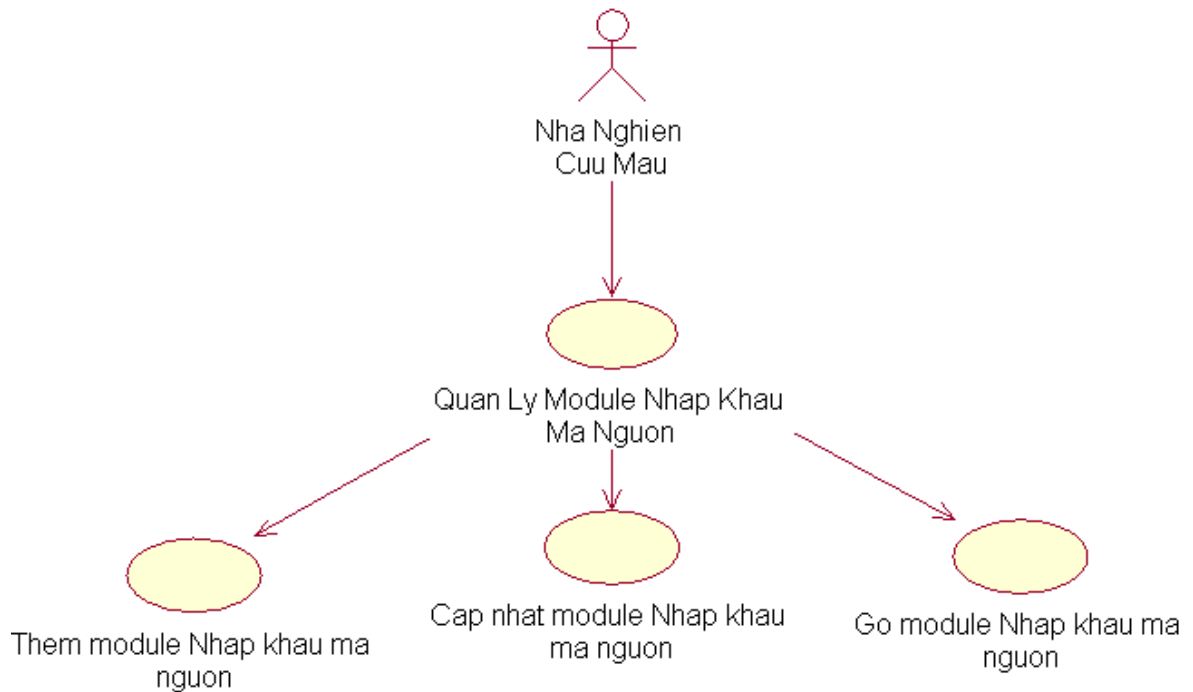


Hình 3-6 Sơ đồ Use-Case Quản lý thể hiện mẫu

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Nghien Cuu Mau	Actor	Nhà nghiên cứu thể hiện mẫu	
2	Quan Ly The Hien Mau	Use-Case	Quản lý thể hiện mẫu	
3	Mo Dac Ta The Hien Mau	Use-Case	Mở tập tin đặc tả của thể hiện mẫu	
4	Luu Dac Ta The Hien Mau	Use-Case	Lưu đặc tả của thể hiện mẫu	
5	Them The Hien Mau Moi	Use-Case	Thêm thể hiện mẫu mới	
6	Them The Hien Mau Tu Mau Da Co	Use-Case	Thêm thể hiện mẫu từ mẫu đã có	
7	Tra Cuu The Hien Mau	Use-Case	Tra cứu thể hiện mẫu	
8	Cap Nhat The Hien Mau	Use-Case	Cập nhật thể hiện mẫu	
9	Xoa The Hien Mau	Use-Case	Xóa thể hiện mẫu	
10	Phat Sinh Ma Nguon	Use-Case	Phát sinh mã nguồn	

Bảng 3-6 Bảng thuyết minh Sơ đồ Use-Case Quản lý thể hiện mẫu

3.1.1.7 Sơ đồ Use-Case Quản lý module nhập vào mã nguồn

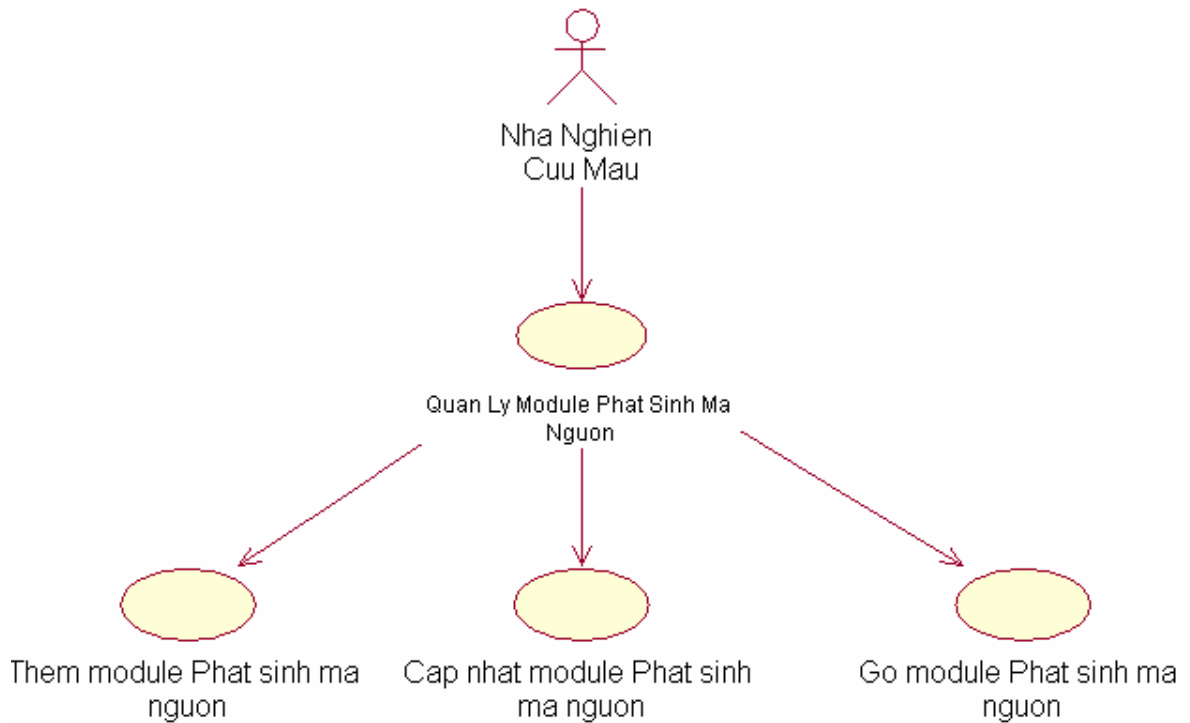


Hình 3-7 Sơ đồ Use-Case Quản lý module nhập vào mã nguồn

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nha Nghien Cuu Mau	Actor	Nhà nghiên cứu thể hiện mẫu	
2	Quan Ly Module Nhap Khau Ma Nguon	Use-Case	Quản lý module nhập mã nguồn	
3	Them Module Nhap Khau Ma Nguon	Use-Case	Thêm module nhập mã nguồn	
4	Cap Nhat Module Nhap Khau Ma Nguon	Use-Case	Cập nhật module nhập mã nguồn	
5	Go Module Nhap Khau Ma Nguon	Use-Case	Gỡ module nhập mã nguồn	

Bảng 3-7 Bảng thuyết minh Sơ đồ Use-Case Quản lý module nhập khẩu mã nguồn

3.1.1.8 Sơ đồ Use-Case Quản lý module phát sinh mã nguồn



Hình 3-8 Sơ đồ Use-Case Quản lý module phát sinh mã nguồn

STT	Tên	Loại	Diễn giải	Ghi chú
1	Nhà Nghiên Cứu Mẫu	Actor	Nhà nghiên cứu thể hiện mẫu	
2	Quản Lý Module Phát Sinh Mã Nguồn	Use-Case	Quản lý module phát sinh mã nguồn	
3	Thêm Module Phát Sinh Mã Nguồn	Use-Case	Thêm module phát sinh mã nguồn	
4	Cập Nhật Module Phát Sinh Mã Nguồn	Use-Case	Cập nhật module phát sinh mã nguồn	
5	Go Module Phát Sinh Mã Nguồn	Use-Case	Gỡ module phát sinh mã nguồn	

Bảng 3-8 Bảng thuyết minh Sơ đồ Use-Case Quản lý module phát sinh mã nguồn

3.1.2 Phân tích các chức năng chính của hệ thống

3.1.2.1 Thiết kế giao diện bằng công cụ đồ họa

Các ngôn ngữ lập trình hiện nay đều hỗ trợ rất tốt chức năng thiết kế giao diện bằng đồ họa. Mọi thao tác trong quá trình thiết kế sẽ được thể hiện trực quan, giúp cho người dùng dễ dàng hơn trong việc đánh giá giao diện của mình.

Để người sử dụng tiếp nhận Hệ thống quản lý mẫu giao diện theo hướng thành phần, ta cũng phải hỗ trợ cho họ có được một môi trường thiết kế giao diện đủ tiện lợi; ngược lại, dù hệ thống có quản lý mẫu rất tốt cũng khó đưa vào sử dụng.

Các yêu cầu về công cụ đồ họa:

- Đảm bảo các chức năng thêm, xóa các điều khiển.
- Cho phép thiết lập các thuộc tính thông dụng của các điều khiển.
- Hỗ trợ các điều khiển chuẩn thường gặp trong hệ thống thông tin.
- Khả năng mở rộng các điều khiển trong tương lai dễ dàng.
- Thân thiện, tự nhiên.

Hướng tiếp cận:

- Phương án 1: Sử dụng một công cụ thiết kế giao diện bằng đồ họa trên thị trường: không khả thi vì thiếu kinh phí.
- Phương án 2: Tự xây dựng một công cụ đồ họa đảm bảo được các điều khiển trên: có thể thực hiện với qui mô ban đầu là các điều khiển chuẩn và đảm bảo khả năng tiến hóa trong tương lai.

Giải pháp: sử dụng phương án 2.

Các vấn đề cần tìm hiểu cho giải pháp:

- Tìm hiểu và nhận diện các điều khiển chuẩn: là những điều khiển được sử dụng rộng rãi, xuất hiện trên nhiều ngôn ngữ lập trình. Với các điều khiển đó, cũng cần phân tích các thuộc tính nào là quan trọng, được sử dụng trong các ngôn ngữ lập trình. Chúng tôi đã đi vào tìm hiểu chủ yếu trên 2 ngôn ngữ lập trình là VB 6.0 của bộ Visual Studio 6.0 và C# của bộ

Visual Studio .Net. Sau đây là danh sách các điều khiển chuẩn đã được tìm kiếm:

STT	Tên	Diễn giải	Ghi chú
1	Label	Nhãn.	
2	Button	Nút chức năng	
3	TextBox	Điều khiển nhập liệu dạng chuỗi	
4	CheckBox	Điều khiển cho phép chọn một/một số/tất cả các lựa chọn	
5	RadioButton	Điều khiển cho phép chọn 1 lựa chọn duy nhất trong nhóm	
6	GroupBox	Nhóm các điều khiển	
7	Grid	Lưới	
8	ListBox	Hộp danh sách đầy đủ, không phân chia cột	
9	ComboBox	Hộp danh sách dạng xổ	
10	ListView	Điều khiển hiển thị danh sách, có thể phân chia cột	
11	TreeView	Điều khiển hiển thị cây	

Bảng 3-9 Danh sách các điều khiển chuẩn trong ngôn ngữ lập trình C# và VB 6.0

- Qua phân tích điều khiển trên, một điều đáng quan tâm chính là khả năng chứa các điều khiển của GroupBox. Điều khiển này có thể chứa bất kỳ các điều khiển chuẩn nào trong 11 điều khiển kể trên, nghĩa là có thể chứa chính nó. Điều lưu ý này cho thấy sau này phải có một bảng thiết kế lớp giải quyết được vấn đề lồng nhau nhiều cấp này.
- Yêu cầu mở rộng điều khiển: Để hệ thống tiến hóa tốt, ta không nên sử dụng trực tiếp các lớp điều khiển trên mà cần thông qua một lớp chung của chúng. Khi cần thêm một điều khiển mới, ta chỉ cần định nghĩa lớp cho điều khiển đó và gắn vào hệ thống các lớp điều khiển hiện hành.

3.1.2.2 Các chức năng Quản lý thành phần, mẫu và thể hiện mẫu

Đối tượng quản lý chính của hệ thống là mẫu và thể hiện mẫu. Để nâng cao khả năng tái sử dụng, hệ thống quản lý từ đơn vị cấu thành của mẫu và thể hiện mẫu đó là thành phần. Việc quản lý thành phần, mẫu, thể hiện mẫu sẽ đem lại những hiệu quả sau:

- Khả năng tái sử dụng không chỉ ở mức độ toàn bộ một mẫu, hay thể hiện mẫu mà còn có thể tái sử dụng từng thành phần của các đối tượng này.
- Mẫu và thể hiện mẫu được xem như là sự lắp ghép của các thành phần khác nhau. Nhờ vậy, có thể hoán vị các thành phần khác nhau để tạo ra nhiều mẫu và thể hiện mẫu khác nhau làm cho khả năng thiết kế mẫu và thể hiện mẫu linh hoạt hơn.

Các yêu cầu:

- Tìm ra cách để đặc tả và lưu trữ cho các đối tượng mà hệ thống quản lý.
- Hỗ trợ các chức năng cơ bản: thêm, xóa, cập nhật, tra cứu các đối tượng trên.
- Khi đang thao tác với mẫu và thể hiện mẫu có thể dễ dàng chuyển sang thao tác với thành phần.

Giải pháp:

- Dùng Xml để đặc tả thành phần, mẫu, thể hiện mẫu.
- Dùng công cụ đồ họa để thiết kế các đối tượng trên.
- Tổ chức các đối tượng thành phần, mẫu, thể hiện mẫu theo cấu trúc phân cấp: thành phần có thông tin về thành phần gốc mà từ đó nó được tạo ra, tương tự cho mẫu, thể hiện mẫu có thông tin về mẫu mà nó được cụ thể hóa.
- Hệ thống hỗ trợ wizard để tra cứu sinh động, nhanh chóng.
- Cho phép thêm một thành phần có sẵn vào mẫu, thể hiện mẫu đang thiết kế.

- Đưa một thành phần của mẫu, thể hiện mẫu vào cơ sở dữ liệu thành phần để tái sử dụng sau này.

3.1.2.3 Các chức năng nhập mã nguồn – phát sinh mã nguồn:

Trong trường hợp, người dùng đã có sẵn một mẫu, hoặc thể hiện mẫu được thiết kế trong một môi trường lập trình cụ thể và có nhu cầu đưa vào hệ thống quản lý. Không thể để người sử dụng phải nhập vào hệ thống bằng cách sử dụng công cụ đồ họa để thiết kế lại từ đầu. Vì vậy, hệ thống cần hỗ trợ chức năng nhập vào hệ thống tập tin mã nguồn của một mẫu hoặc thể hiện mẫu

Mẫu và thể hiện mẫu là một kiểu dữ liệu đặc biệt, cần phải có một cách đặc tả để lưu trữ. Do đó, cần phải có sự chuyển đổi từ đặc tả sang một ngôn ngữ lập trình cụ thể khi người sử dụng có nhu cầu lấy mẫu, thể hiện mẫu ra để sử dụng.

Hướng tiếp cận:

- Phương án 1: Sử dụng các công cụ có sẵn hỗ trợ việc nhập (import) và phát sinh (generate) mã nguồn cho mẫu thể hiện mẫu: không khả thi vì thiếu kinh phí.
- Phương án 2: tự xây dựng hai chức năng trên, có thể thực hiện ở qui mô ban đầu trên một số ngôn ngữ lập trình và môi trường lập trình và đảm bảo khả năng tiến hóa trong tương lai.

Giải pháp:

- Chọn lựa phương án 2.
- Qua khảo sát: quyết định hệ thống sẽ hỗ trợ hai chức năng trên cho các ngôn ngữ lập trình sau:
 - o Ngôn ngữ VB6.0 trong môi trường Visual Studio 6.0. Lý do: Visual Studio 6.0 là một môi trường lập trình được sử dụng phổ biến. VB6.0 là một ngôn ngữ thường sử dụng trong các ứng dụng Hệ thống thông tin khi môi trường được lựa chọn là Visual Studio 6.0.

- Ngôn ngữ C# trong môi trường Visual Studio .NET. Lý do: Visual Studio .NET là một môi trường lập trình hiện đại, hỗ trợ mạnh các tính năng, được sử dụng rất phổ biến hiện nay. Trong đó C# là một trong những ngôn ngữ lập trình được sử dụng nhiều nhất.
- Để đảm bảo khả năng tiến hóa: mỗi chức năng nhập mã nguồn, phát sinh mã nguồn cho một ngôn ngữ lập trình không thể áp dụng cứng nhắc vào hệ thống ban đầu mà phải được module hóa. Khi có nhu cầu sẽ được plug – in vào hệ thống mà không cần phải biên dịch lại chương trình.

3.1.2.4 Chức năng xử lý plug – in các module

Như đã nói ở trên, hệ thống cần chức năng xử lý plug – in các module nhập mã nguồn của mẫu, thể hiện mẫu vào hệ thống và phát sinh mã nguồn cho mẫu, thể hiện mẫu.

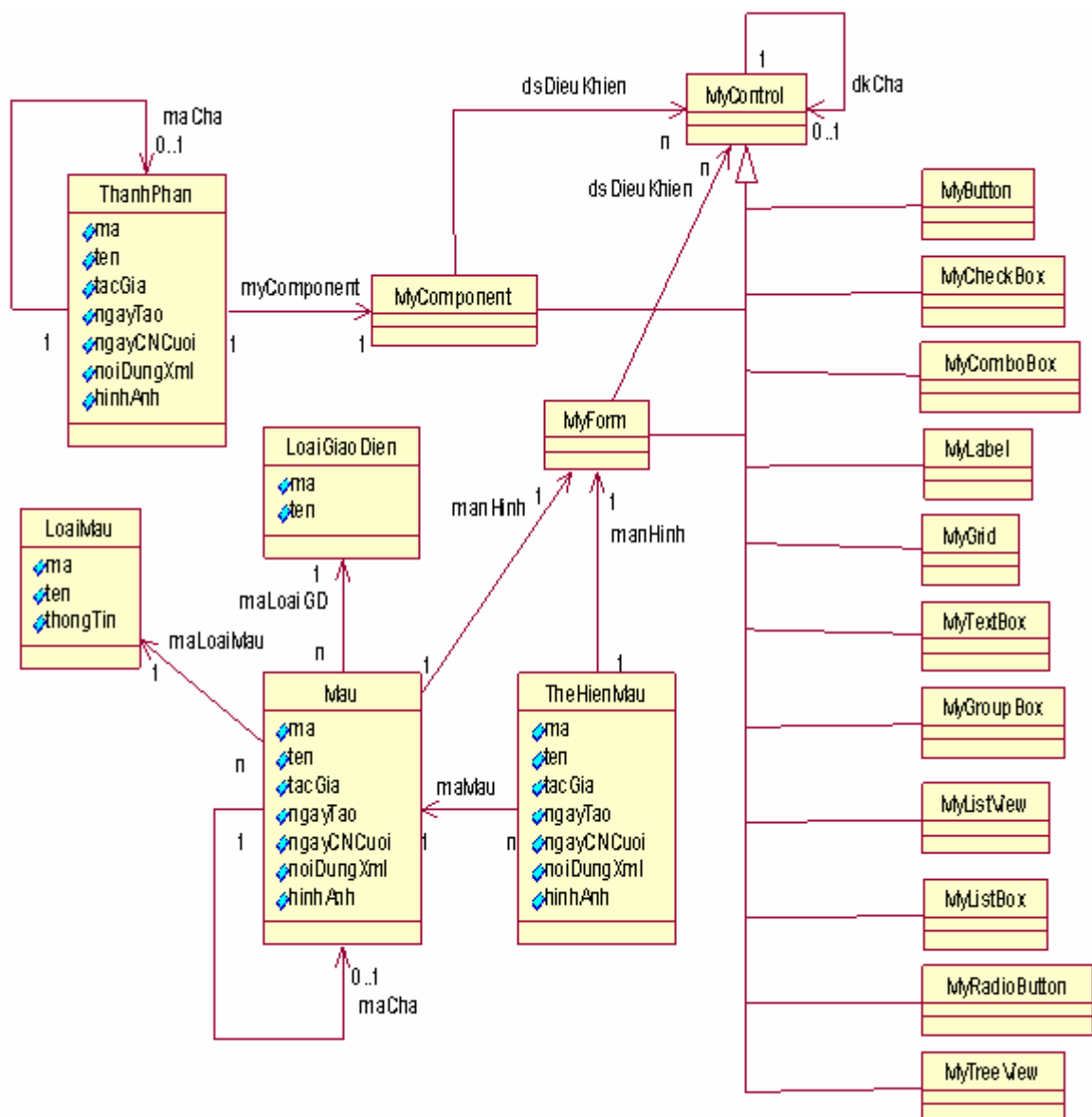
Các yêu cầu:

- Cho phép plug-in module tại thời điểm thực thi và không phải biên dịch lại chương trình.

Giải pháp:

- Thiết kế các module này có kèm theo một lớp đối tượng đóng vai trò giao tiếp giữa module và hệ thống quản lý mẫu giao diện theo ý tưởng của mẫu Proxy như đã trình bày ở mục **Error! Reference source not found.**
- Tận dụng khả năng load tập tin .dll tại thời điểm thực thi mà môi trường lập trình có hỗ trợ.

3.1.3 Sơ đồ lớp mức phân tích

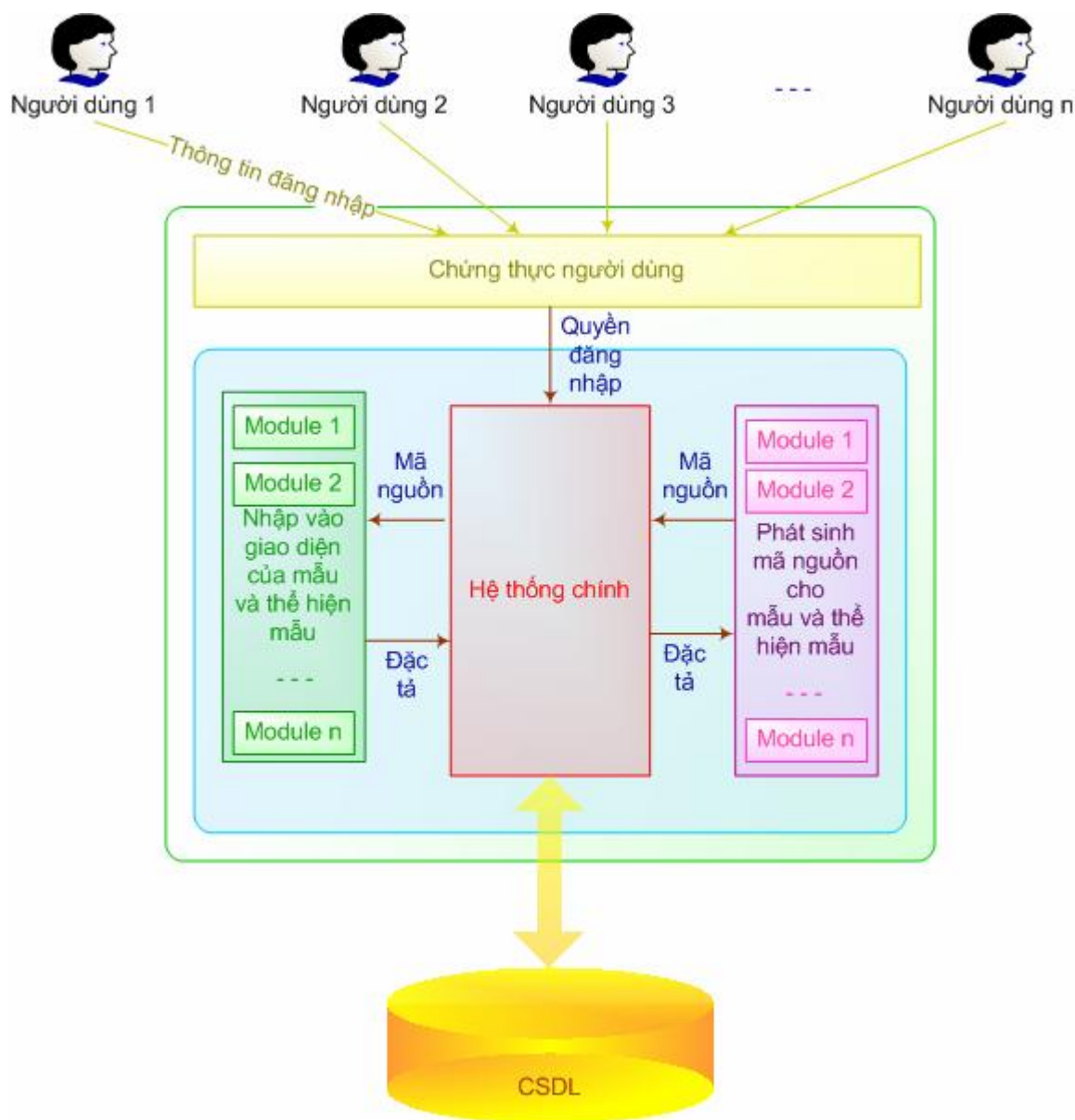


Hình 3-9 Sơ đồ lớp mức phân tích

3.2 Thiết kế

3.2.1 Kiến trúc tổng thể

3.2.1.1 Kiến trúc triển khai

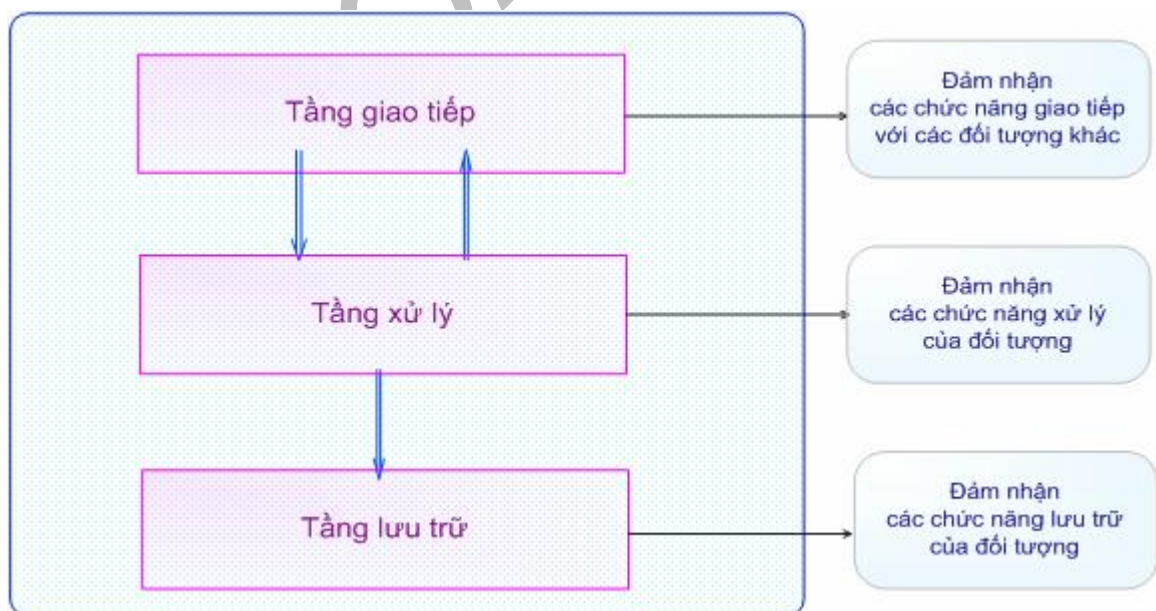


Hình 3-10 Kiến trúc triển khai

Hệ thống Quản lý mẫu giao diện bao gồm :

- Chứng thực người dùng : với 3 quyền đăng nhập Nhà quản trị, Nhà nghiên cứu mẫu và Người sử dụng.
- Hệ thống chính: đảm nhận các chức năng chính là quản lý hệ thống, quản lý thành phần, quản lý mẫu, quản lý thể hiện mẫu, quản lý các module.
- Các module nhập vào giao diện của mẫu và thể hiện mẫu:
 - o Nhận đặc tả Xml của mẫu hay thể hiện mẫu từ hệ thống.
 - o Tiến hành chuyển đặc tả Xml thành một tập tin mã nguồn trên một ngôn ngữ được lựa chọn. Tập tin này có thể gắn vào một ứng dụng. Khi thực thi ứng dụng, người dùng sẽ có một giao diện của mẫu hay thể hiện mẫu.
 - o Tạo ra đính kèm tập tin hướng dẫn sử dụng nếu có người dùng có yêu cầu.
- Các module phát sinh mã nguồn cho mẫu và thể hiện mẫu:
 - o Nhận tập tin mã nguồn của một giao diện
 - o Tiến hành chuyển mã nguồn thành đặc tả Xml của mẫu hay thể hiện mẫu (tùy theo yêu cầu người dùng) để đưa vào hệ thống.

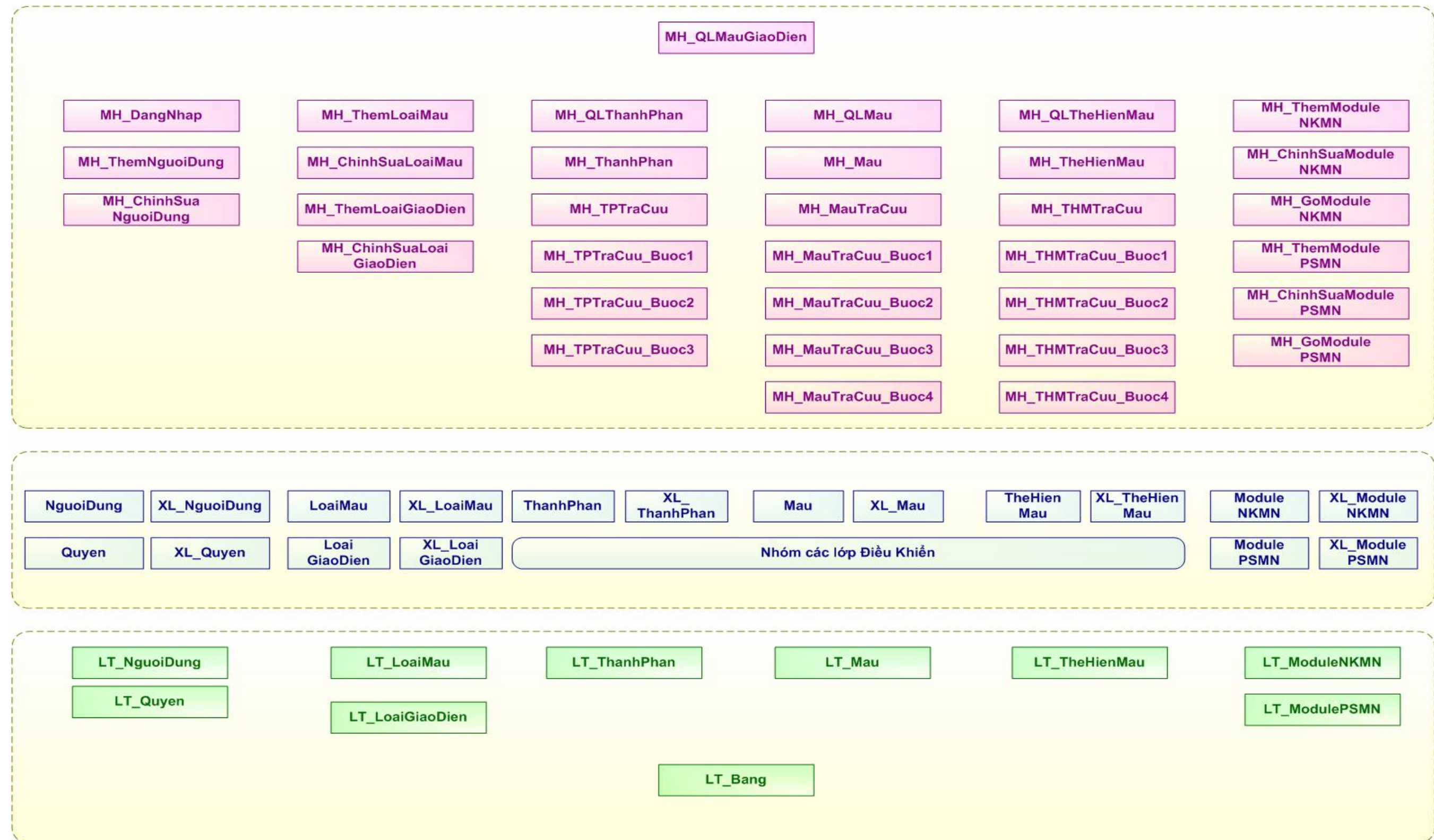
Hệ thống được triển khai theo mô hình 3 tầng, chức năng của mỗi tầng được mô tả như sau:



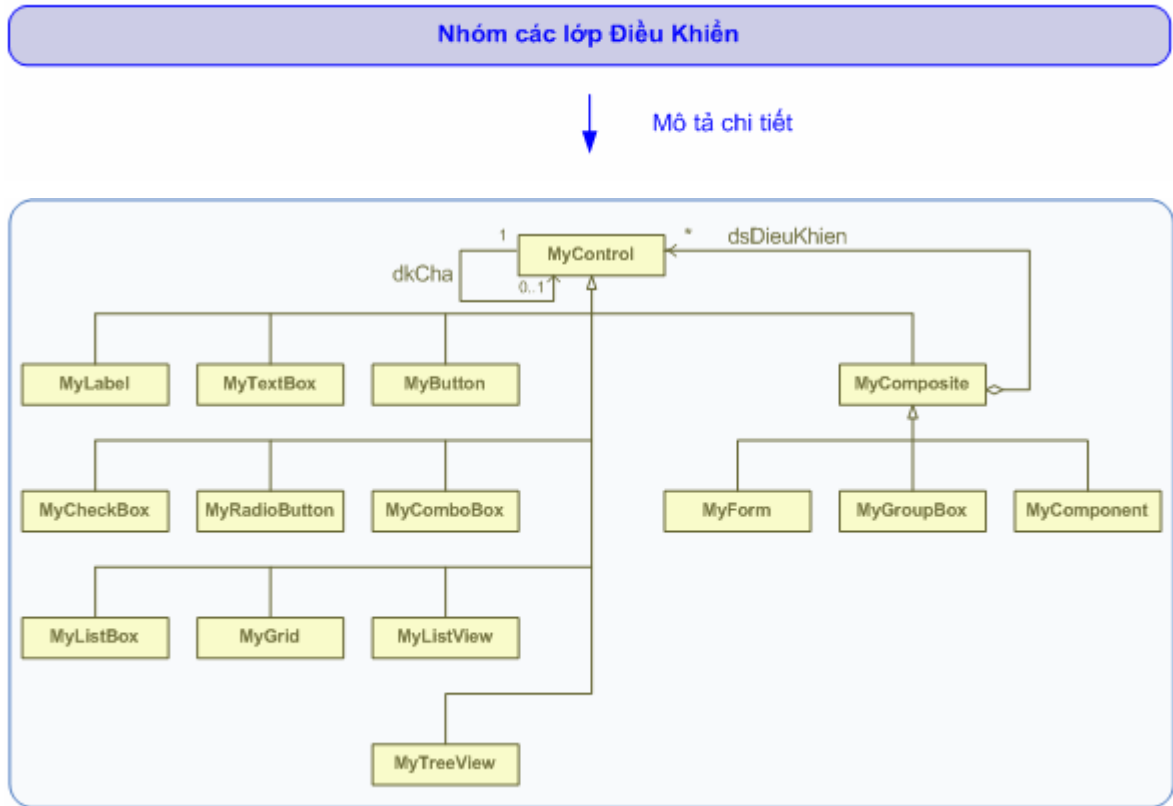
Hình 3-11 Mô hình ba tầng được triển khai

3.2.1.2 Kiến trúc logic

- Trên hệ thống chính:



Hình 3-12 Kiến trúc logic trên hệ thống chính



Hình 3-13 Mô tả chi tiết nhóm các lớp Điều Khiển

Ghi chú: đối với các đối tượng: MyComposite, MyComponent, MyForm, MyControl.. thuộc nhóm các lớp điều khiển phụ vụ cho các chức năng đồ họa, tạo đặc tả, phát sinh mã nguồn ... không có nhu cầu giao tiếp với cơ sở dữ liệu nên không được thiết kế theo kiến trúc 3 tầng.

Ý nghĩa của sơ đồ kiến trúc logic:

Hệ thống được thiết kế theo mô hình 3 tầng. Trong đó:

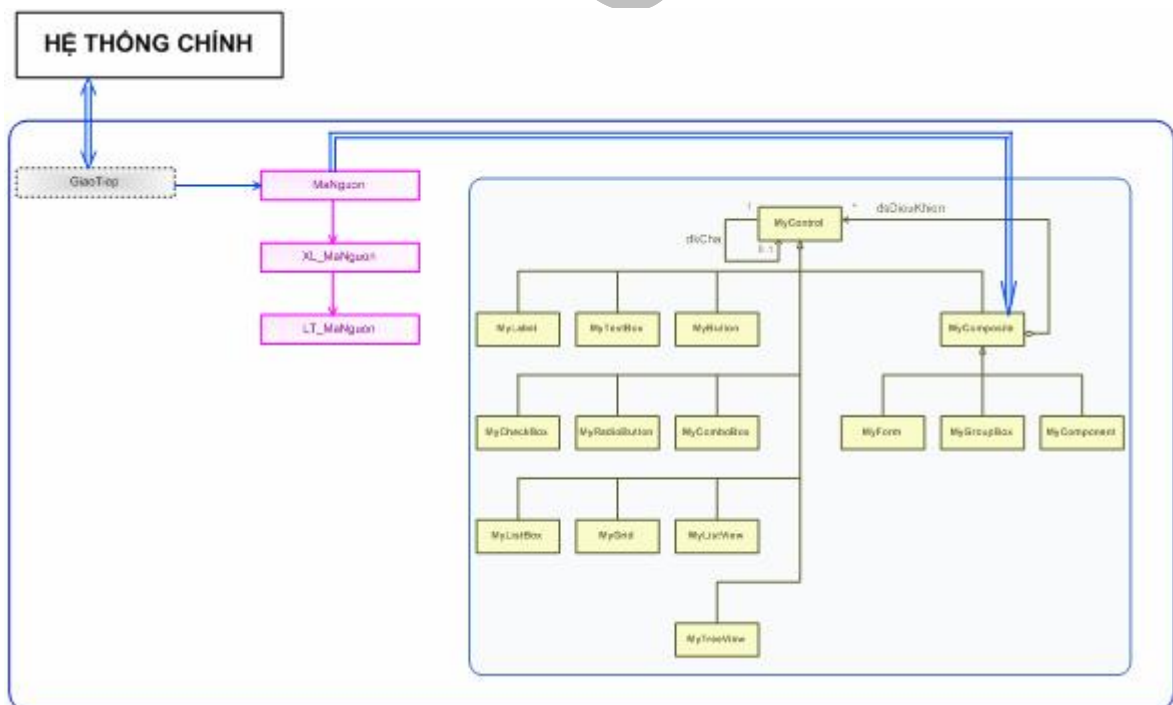
- Tầng giao tiếp bao gồm các màn hình thực hiện chức năng giao tiếp với người dùng.
- Tầng xử lý bao gồm các lớp đối tượng đảm nhận vai trò xử lý cho các chức năng mà hệ thống có hỗ trợ.
- Tầng lưu trữ bao gồm các lớp đối tượng đảm nhận chức năng thực hiện các thao tác với cơ sở dữ liệu.

Các bước tiến hành khi người sử dụng yêu cầu thực hiện một chức năng trong hệ thống:

- Người sử dụng thao tác với các đối tượng màn hình giao diện thuộc tầng giao tiếp của của hệ thống. Các đối tượng này có chức năng nhận những dữ liệu và thao tác của người sử dụng.
- Khi có nhu cầu thực hiện xử lý, đối tượng màn hình sẽ gọi các đối tượng ở tầng xử lý thực hiện xử lý.
- Nếu các đối tượng ở tầng xử lý có nhu cầu lấy dữ liệu dưới cơ sở dữ liệu. Các đối tượng này sẽ gọi đối tượng tương ứng ở tầng lưu trữ. Sau đó các đối tượng ở tầng lưu trữ sẽ tiến hành lấy dữ liệu trong cơ sở dữ liệu và trả về cho các đối tượng ở tầng xử lý.

Các đối tượng ở tầng xử lý sẽ trả dữ liệu cho các đối tượng màn hình ở tầng giao tiếp. Các đối tượng màn hình sẽ lấy dữ liệu, đổ vào các điều khiển và hiển thị cho người dùng.

Trên các module nhập vào giao diện của mẫu và thể hiện mẫu và các module phát sinh mã nguồn cho mẫu và thể hiện mẫu: các module này đều có cùng sơ đồ logic, chỉ khác nhau ở nội dung xử lý bên trong.



Hình 3-14 Kiến trúc logic trên các module nhập và các module phát sinh mã nguồn

3.2.2 Sơ đồ tuần tự - sơ đồ cộng tác cho một số xử lý chính:

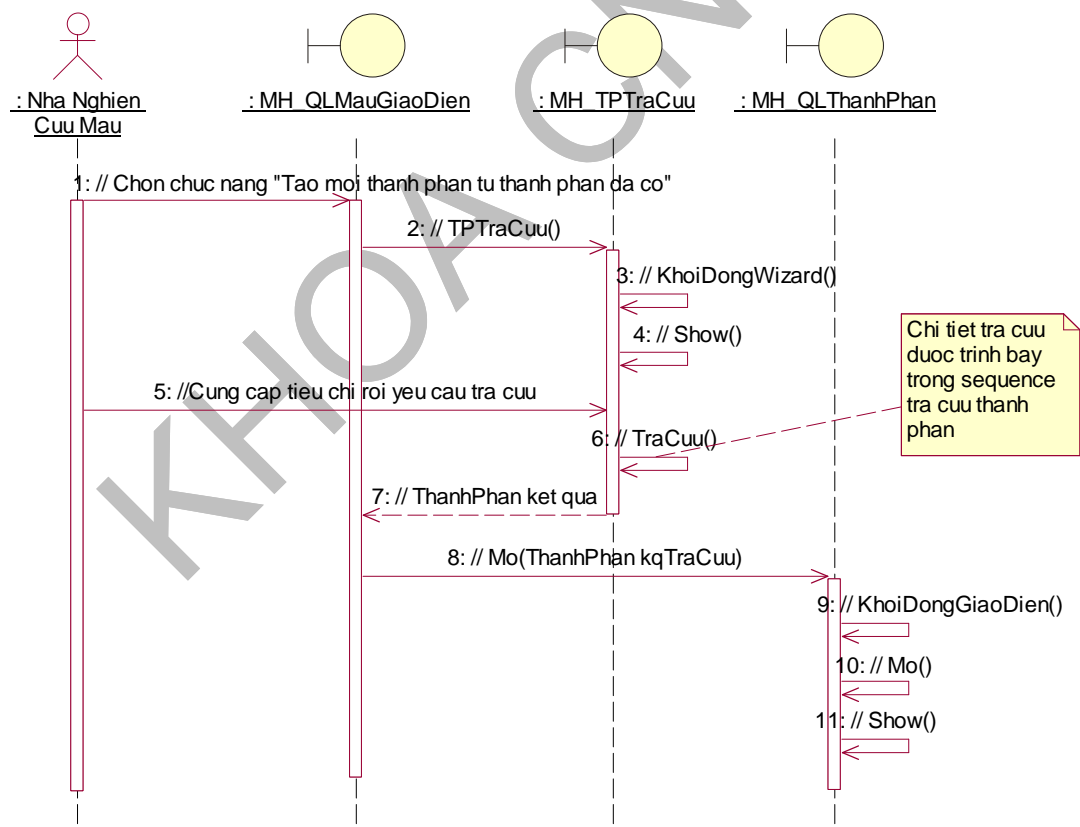
3.2.2.1 Tra cứu thành phần

Chức năng tra cứu thành phần được thực hiện thông qua một wizard tra cứu giúp người dùng thực hiện công việc một cách dễ dàng.

Wizard bao gồm một nhóm các màn hình:

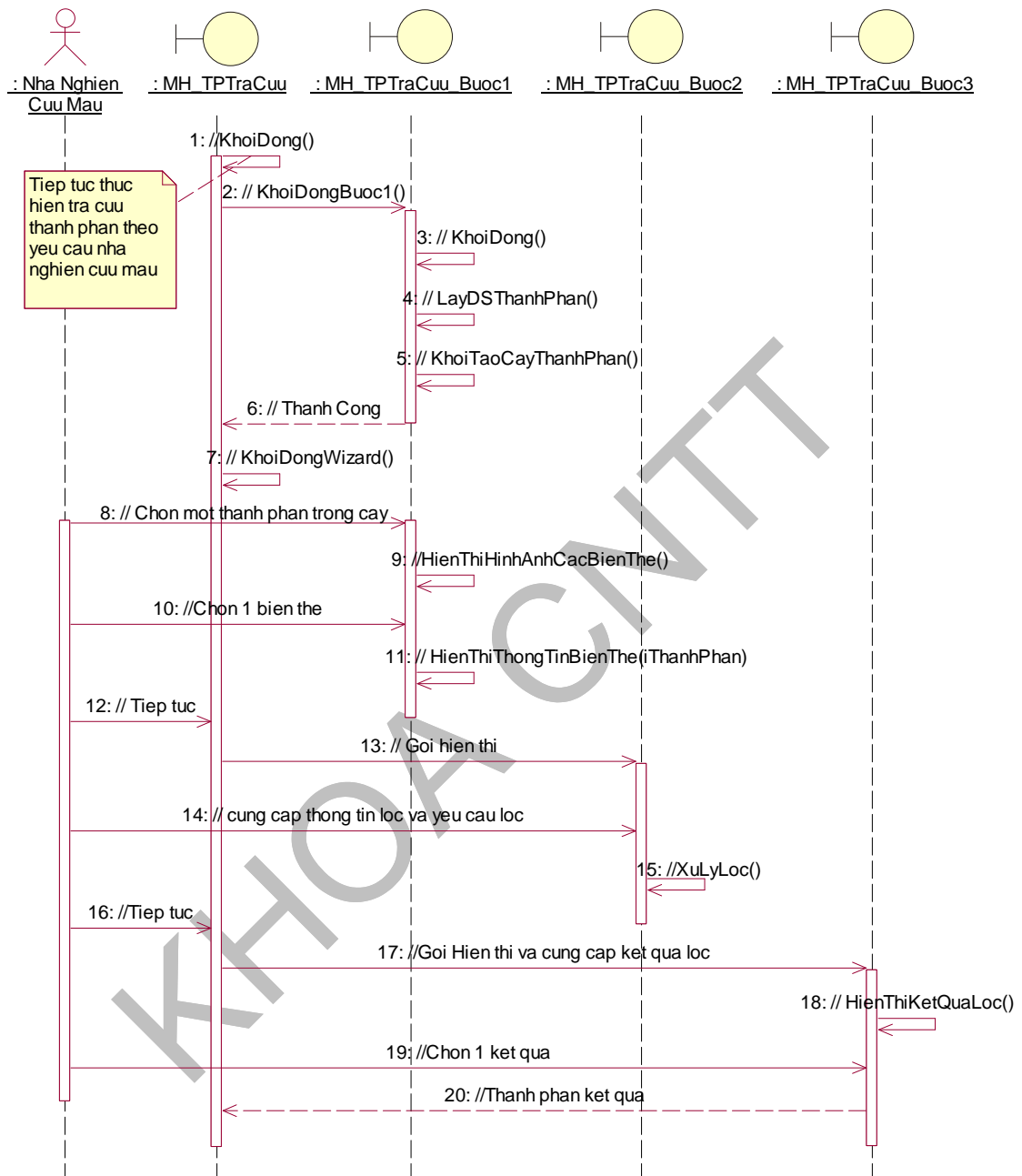
- MH_TPTraCuu (là màn hình chính trong nhóm).
- MH_TPTraCuu_Buoc1 (tương ứng với bước 1 trong wizard tra cứu).
- MH_TPTraCuu_Buoc2 (tương ứng với bước 2 trong wizard tra cứu).
- MH_TPTraCuu_Buoc3 (tương ứng với bước 3 trong wizard tra cứu).

Kết quả tra cứu sẽ được mở trong màn hình MH_QLThanhPhan để người dùng có thể thực hiện các chức năng quản lý thành phần như cập nhật, xóa, tạo biến thể.



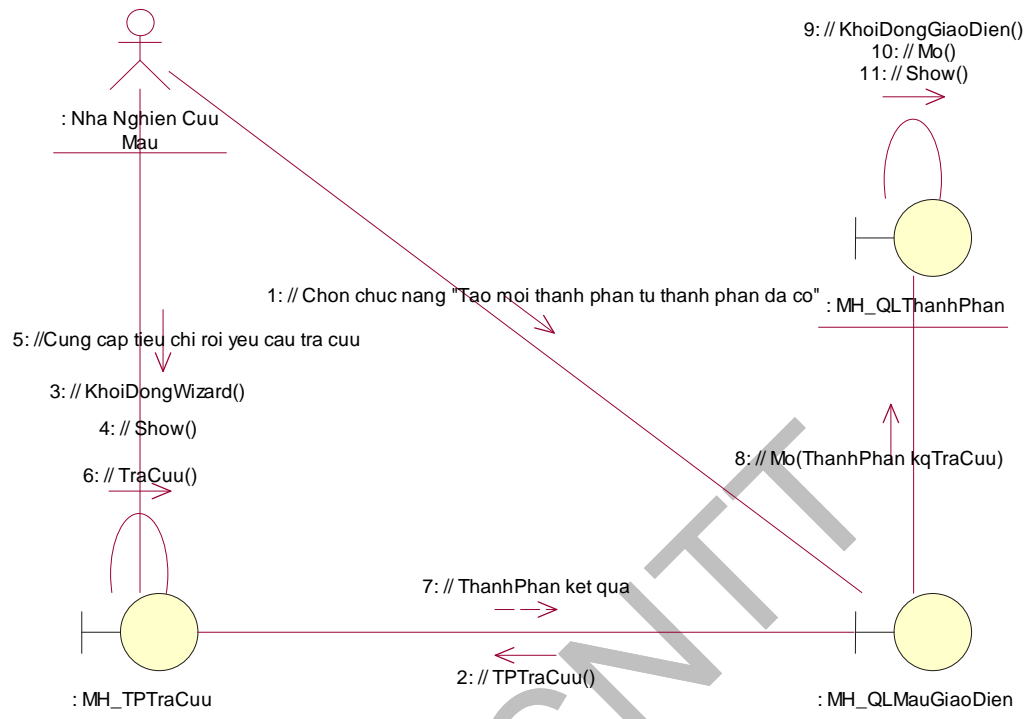
Hình 3-15 Sơ đồ tuần tự của xử lý tra cứu thành phần

Sơ đồ tuần tự này sẽ mô tả rõ ràng các bước tiến hành tra cứu (chưa được thể hiện trong sơ đồ ở trên).

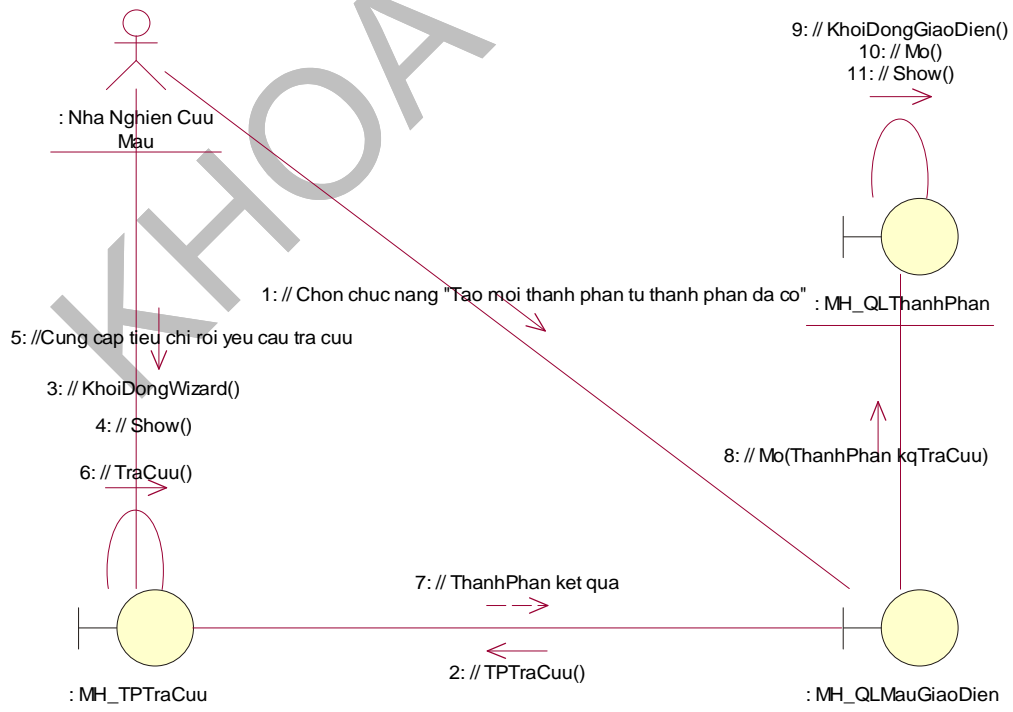


Hình 3-16 Sơ đồ tuần tự của xử lý chi tiết tra cứu thành phần

Dưới đây là sơ đồ cộng tác tương ứng với 2 sơ đồ tuần tự của xử lý tra cứu thành phần.



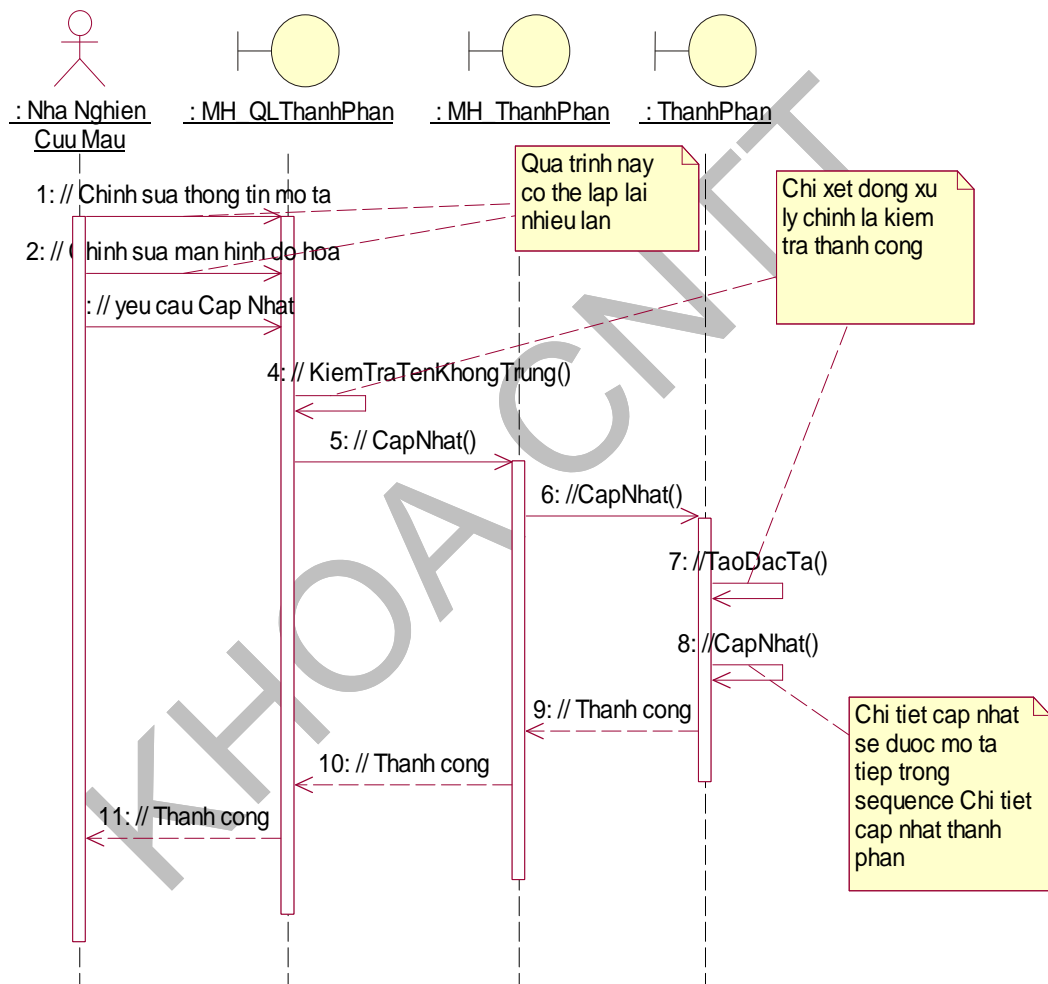
Hình 3-17 Sơ đồ cộng tác xử lý tra cứu thành phần



Hình 3-18 Sơ đồ cộng tác chi tiết xử lý tra cứu thành phần

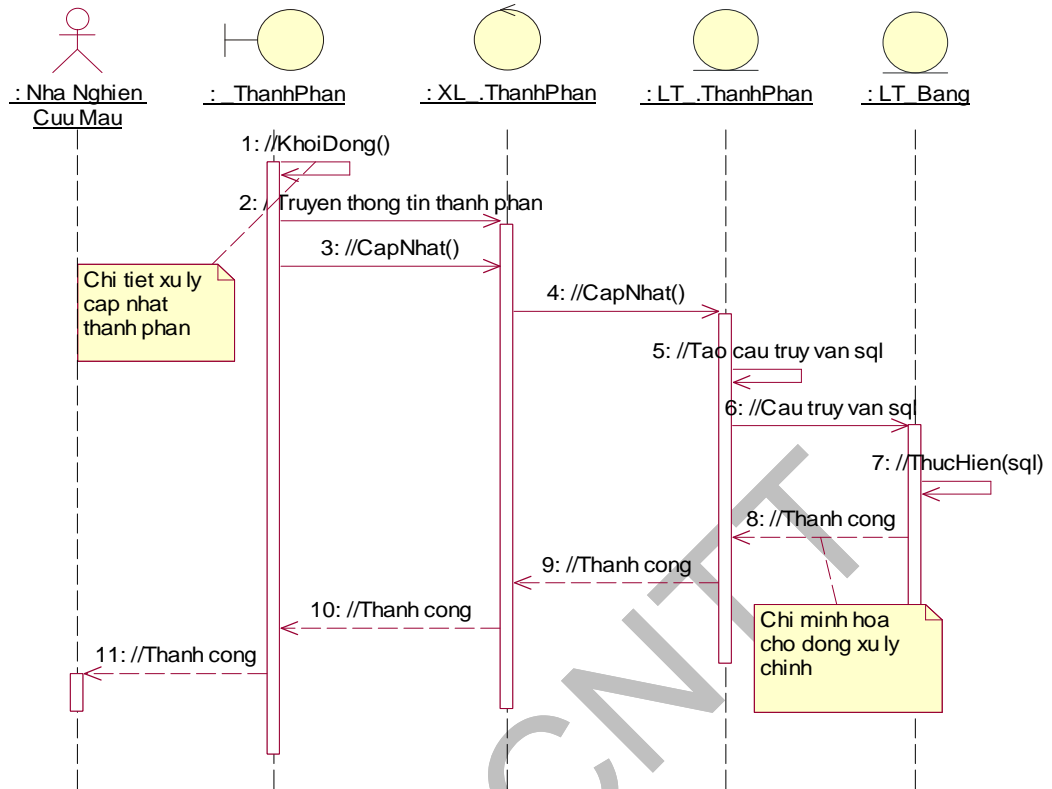
3.2.2.2 Cập nhật thành phần

Xử lý cập nhật thành phần được thực hiện khi người dùng đang sử dụng màn hình MH_QLThanhPhan (màn hình quản lý thành phần). Sau khi chỉnh sửa các thông tin mô tả của thành phần hay/và giao diện của thành phần, người dùng sẽ yêu cầu cập nhật. Hệ thống phải kiểm tra thao tác cập nhật có hợp lệ hay không (người dùng chính là tác giả của thành phần và tên thành phần không được trùng với một thành phần đã có trong hệ thống) trước khi tiến hành cập nhật.



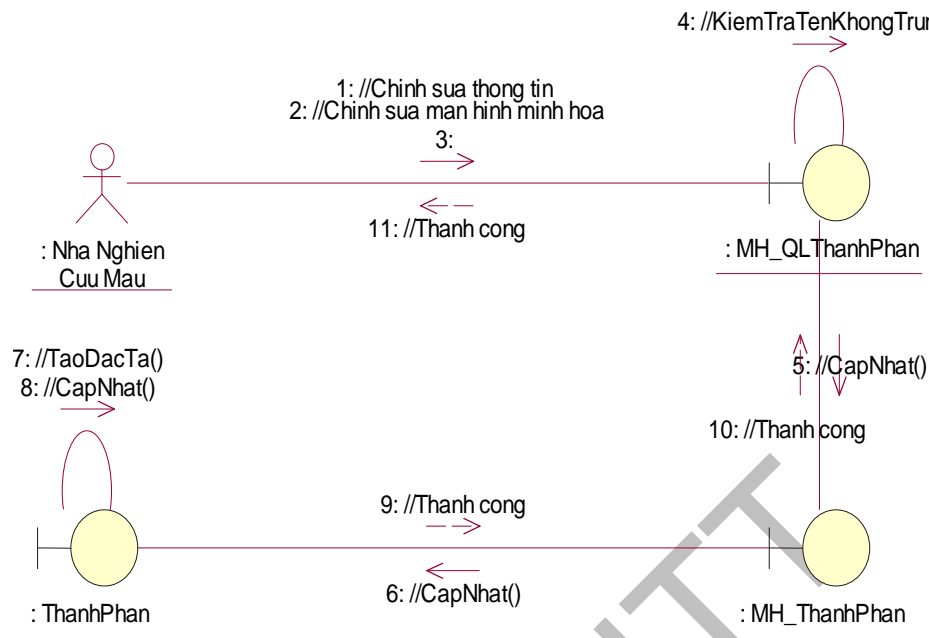
Hình 3-19 Sơ đồ tuần tự xử lý cập nhật thành phần

Đây là sơ đồ tuần tự mô tả chi tiết quá trình cập nhật thành phần theo mô hình 3 lớp.

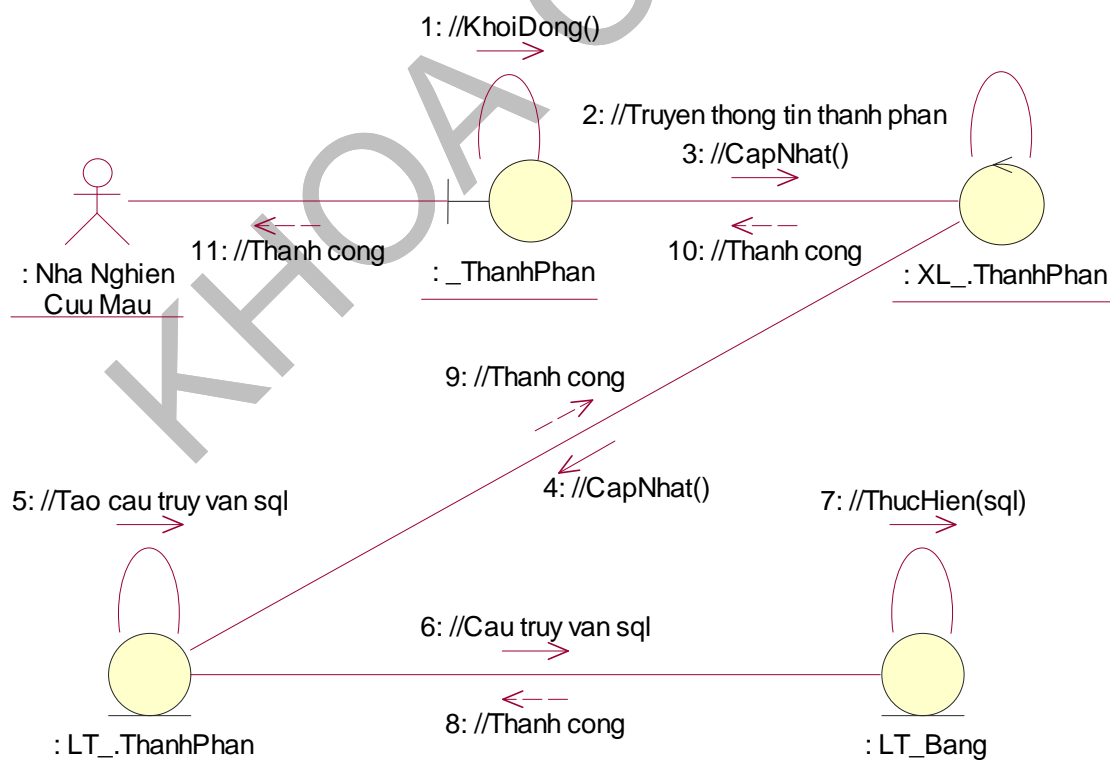


Hình 3-20 Sơ đồ tuần tự chi tiết cập nhật thành phần

Dưới đây là 2 sơ đồ cộng tác tương ứng với 2 sơ đồ tuần tự của xử lý cập nhật thành phần.



Hình 3-21 Sơ đồ cộng tác xử lý cập nhật thành phần



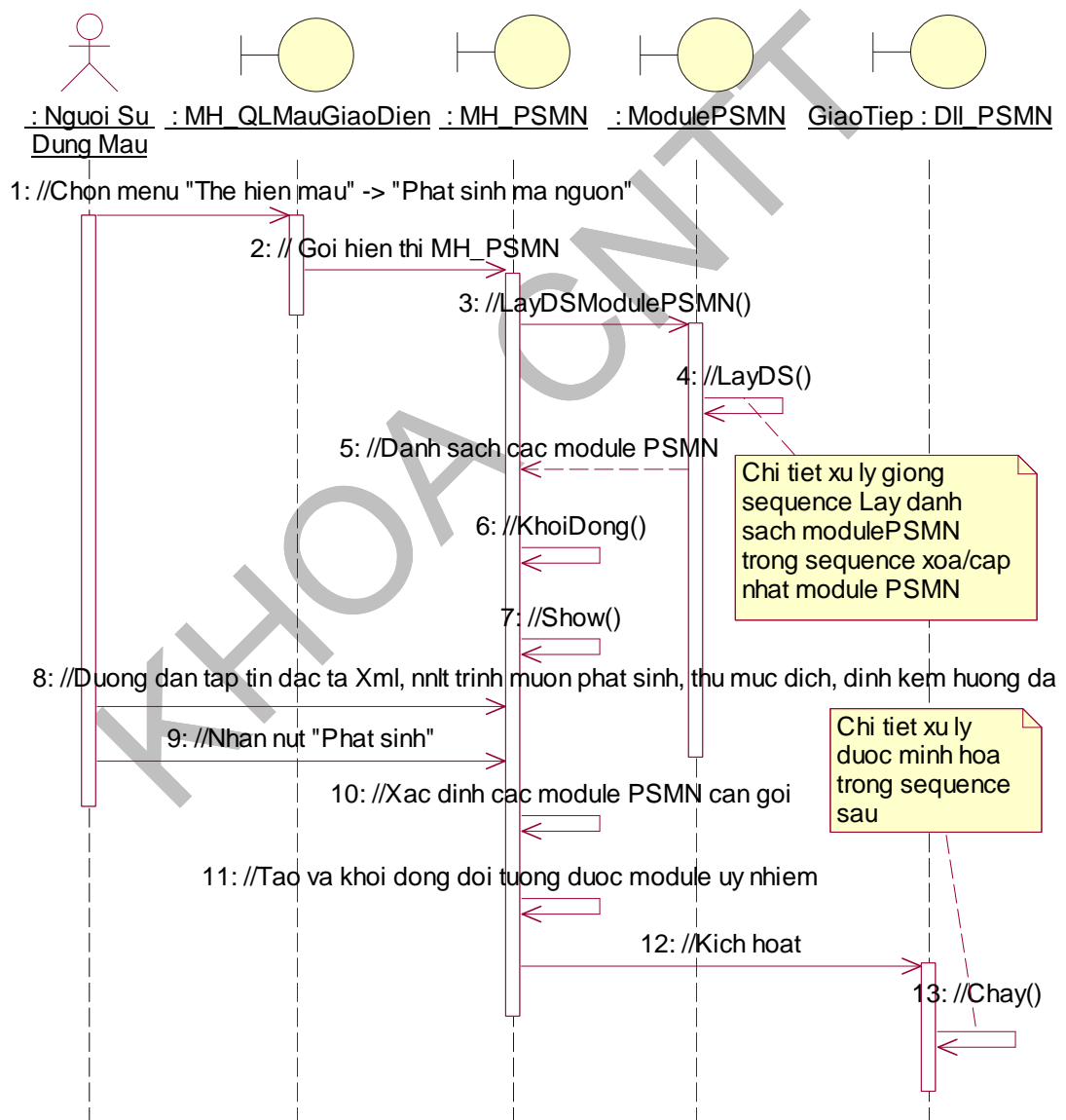
Hình 3-22 Sơ đồ cộng tác chi tiết xử lý cập nhật thành phần

3.2.2.3 Phát sinh mã nguồn cho thể hiện mẫu được gọi từ menu chính

Khi sử dụng hệ thống, người dùng có thể yêu cầu phát sinh mã nguồn của một thể hiện mẫu trong 2 trường hợp:

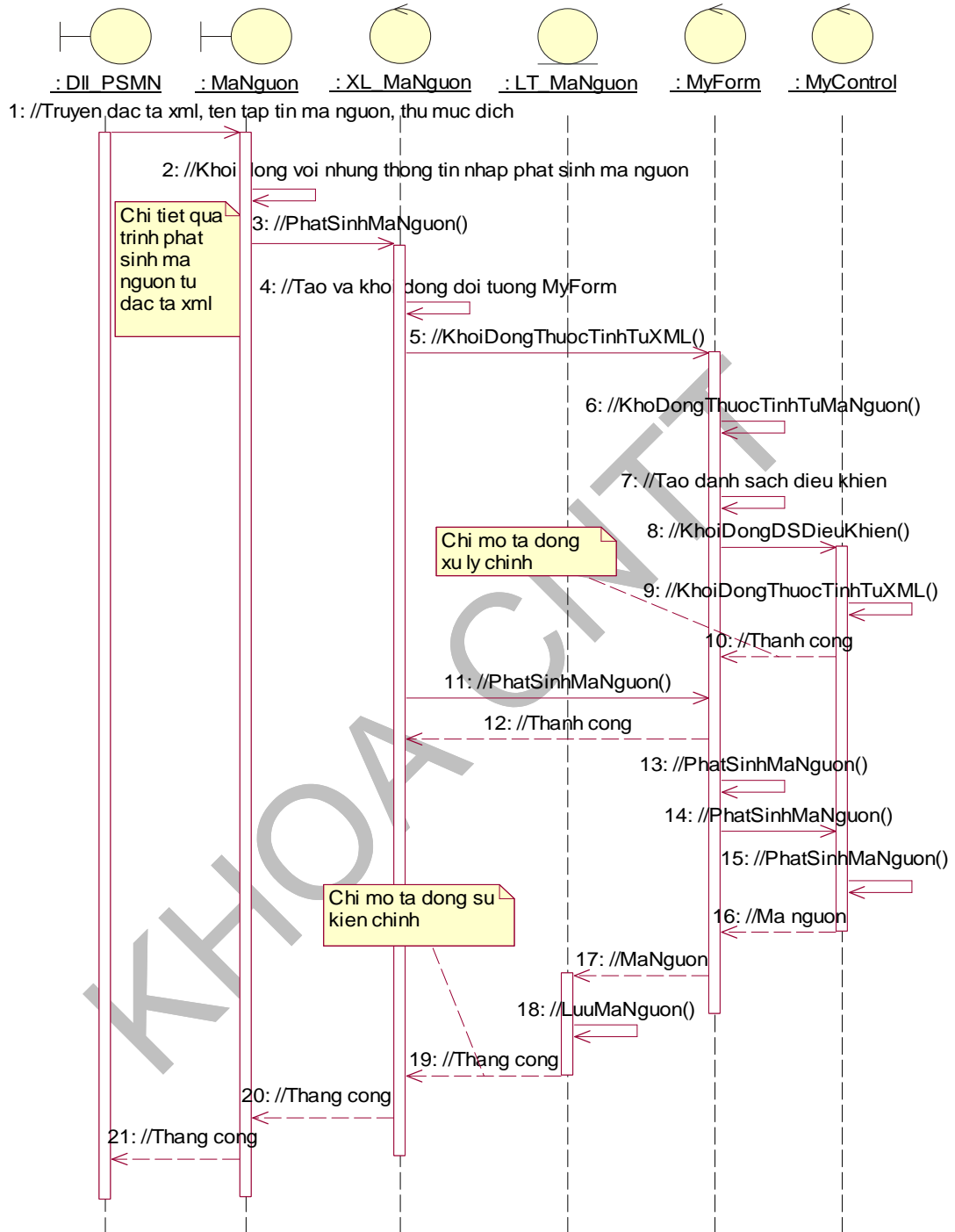
- Dùng MH_QLMauGiaoDien (màn hình chính của hệ thống) để yêu cầu phát sinh mã nguồn từ một đặc tả thể hiện mẫu (tập tin xml).
- Đang quản lý thể hiện mẫu bằng màn hình MH_QLTheHienMau (màn hình quản lý thể hiện mẫu).

Dưới đây là sơ đồ tuần tự cho cách thứ 1.



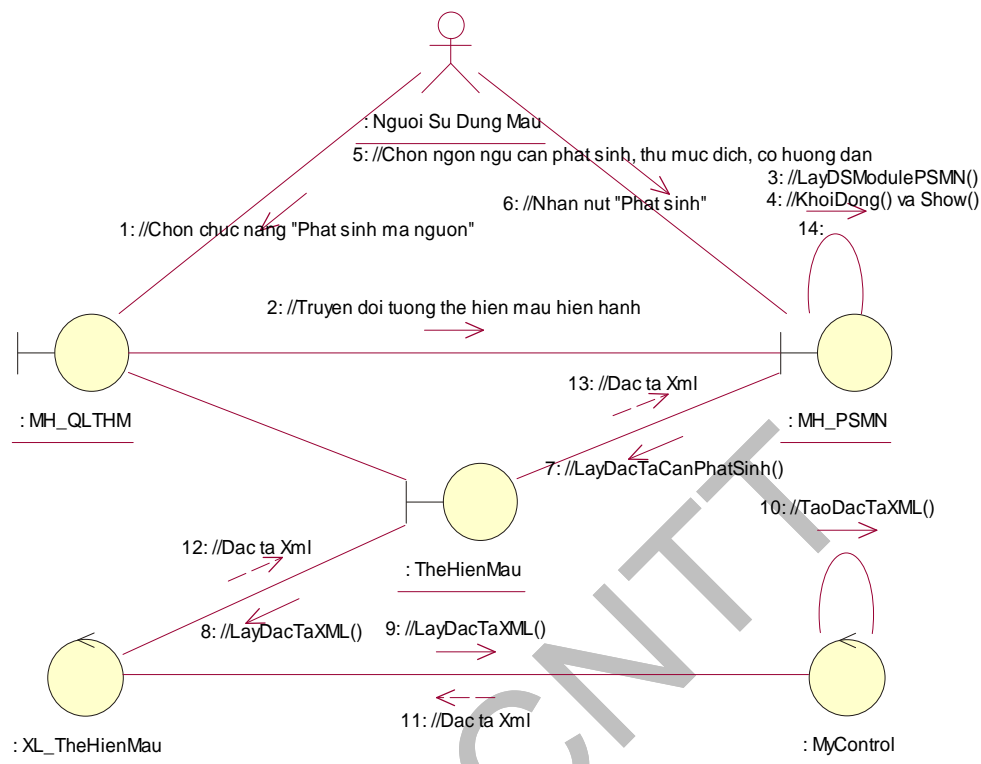
Hình 3-23 Sơ đồ tuần tự xử lý phát sinh mã nguồn gọi từ menu chính

Đây là sơ đồ tuần tự mô tả cách xử lý phát sinh mã nguồn trên module phát sinh mã nguồn.

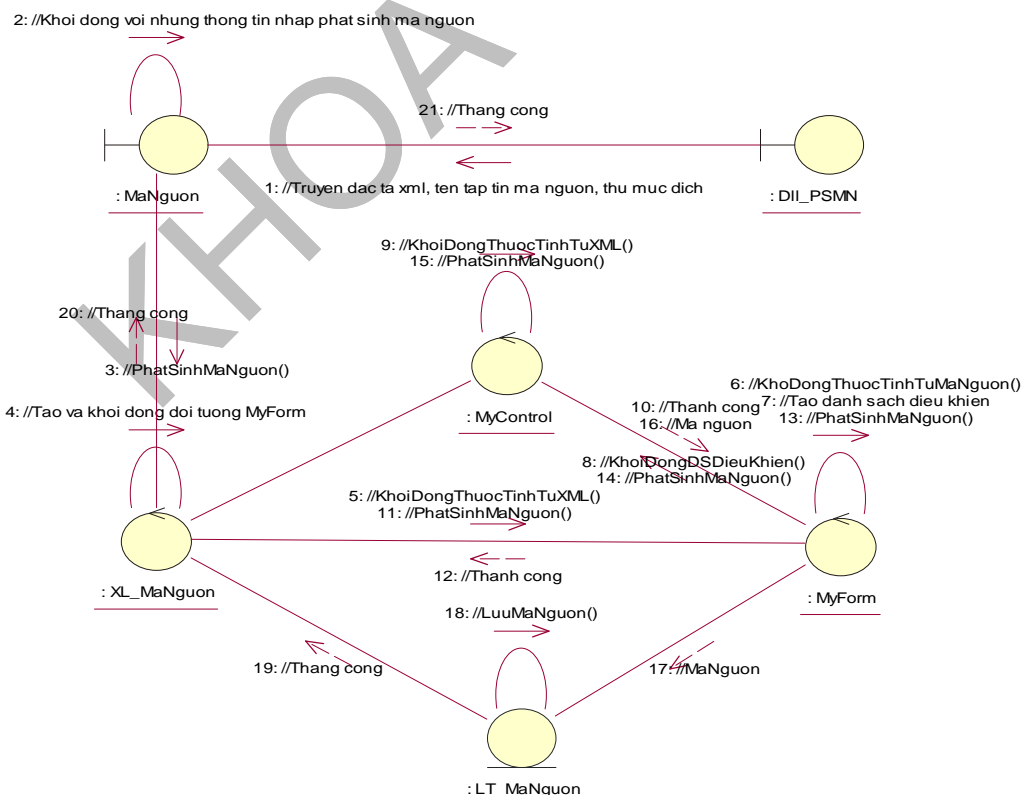


Hình 3-24 Sơ đồ tuần tự chi tiết xử lý phát sinh mã nguồn cho thẻ hiện mẫu

Dưới đây là 2 sơ đồ cộng tác của xử lý phát sinh mã nguồn cho thể hiện mẫu dựa vào tập tin đặc tả xml.



Hình 3-25 Sơ đồ cộng tác xử lý phát sinh mã nguồn cho thể hiện mẫu gọi từ menu chính

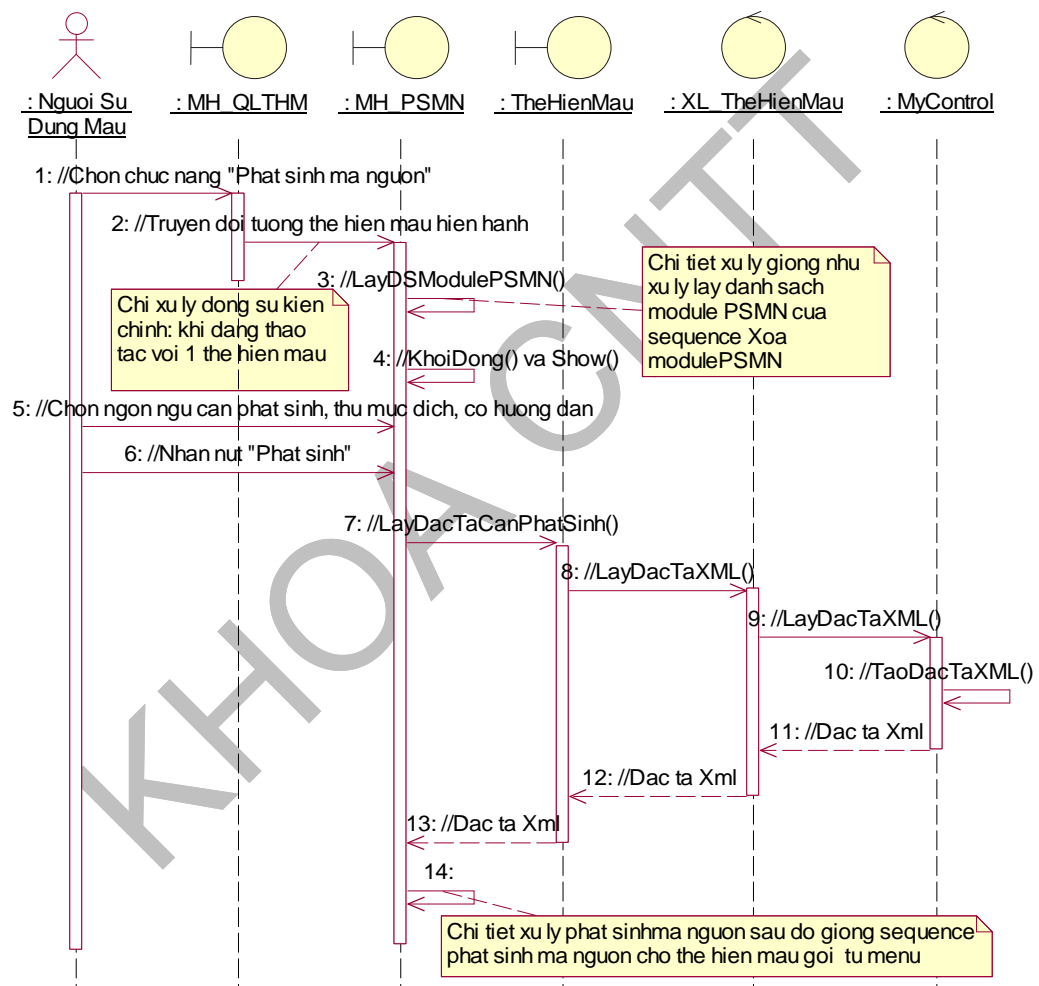


Hình 3-26 Sơ đồ cộng tác chi tiết xử lý phát sinh mã nguồn trong module

3.2.2.4 Gọi phát sinh mã nguồn cho một thể hiện mẫu đang thiết kế trong hệ thống

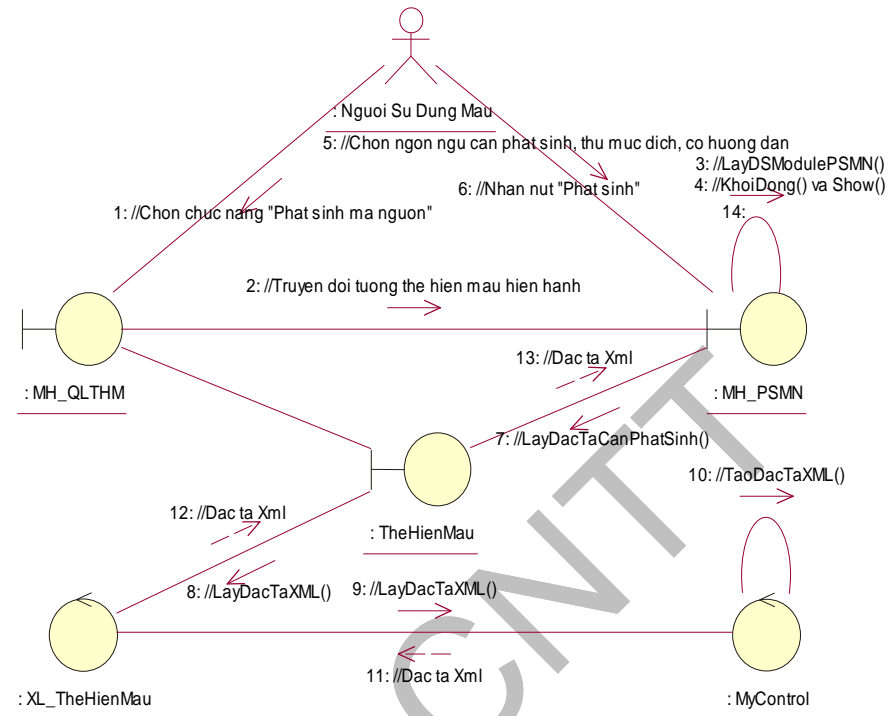
Dưới đây là sơ đồ tuần tự cho cách phát sinh mã nguồn của một thể hiện mẫu đang được quản lý bằng màn hình MH_QLTheHienMau (màn hình quản lý thể hiện mẫu).

Quá trình phát sinh mã nguồn khi gọi module phát sinh mã nguồn cũng tương tự như xử lý Gọi phát sinh mã nguồn cho một thể hiện mẫu từ menu chính.



Hình 3-27 Sơ đồ tuần tự cho xử lý phát sinh mã nguồn cho thể hiện mẫu đang thiết kế

Tương ứng với sơ đồ tuần tự vừa rồi là một sơ đồ cộng tác của xử lý phát sinh mã nguồn cho thể hiện mẫu trong màn hình quản lý.

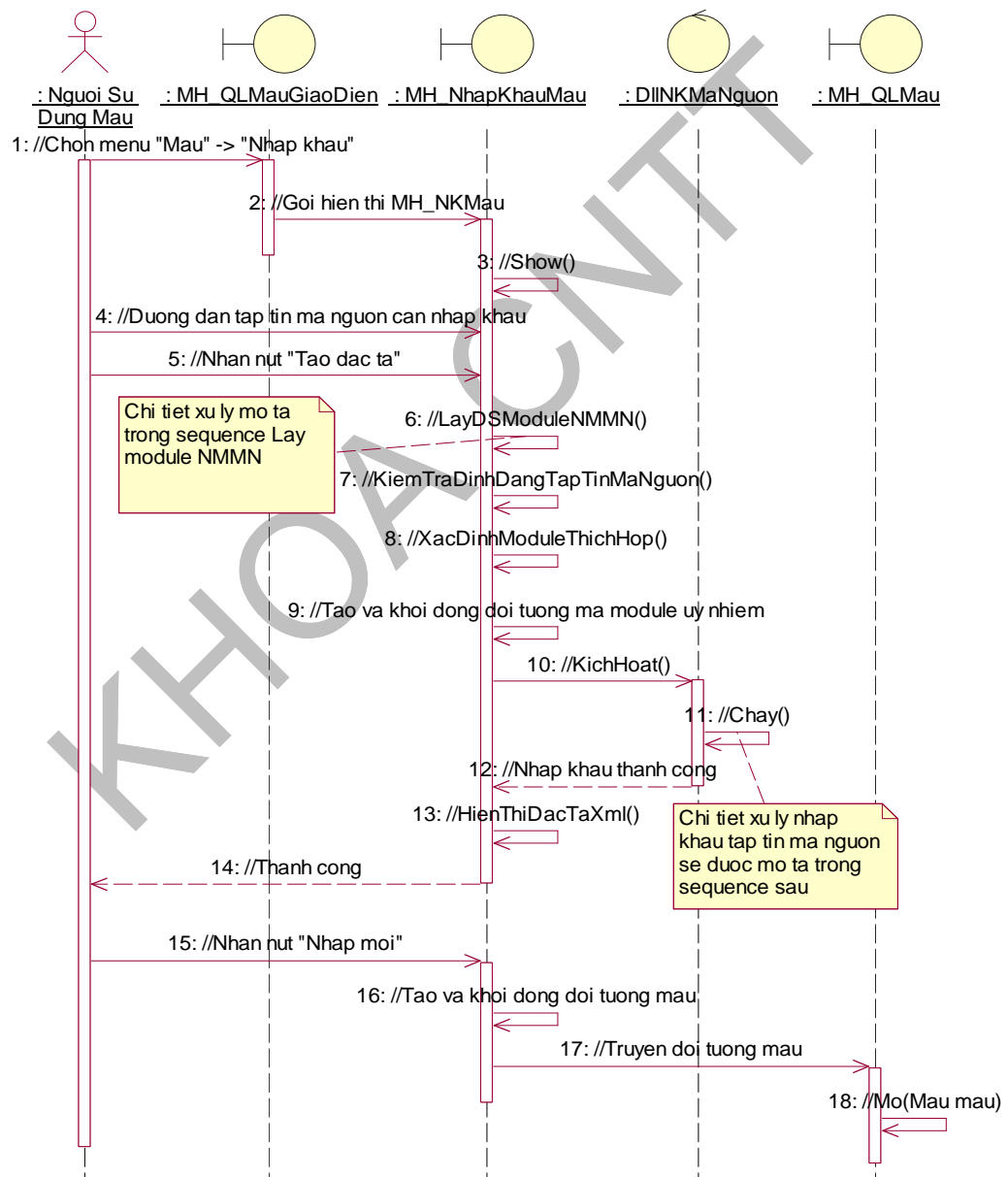


Hình 3-28 Sơ đồ cộng tác xử lý phát sinh mã nguồn cho thể hiện mẫu đang thiết kế

3.2.2.5 Nhập mã nguồn cho mẫu

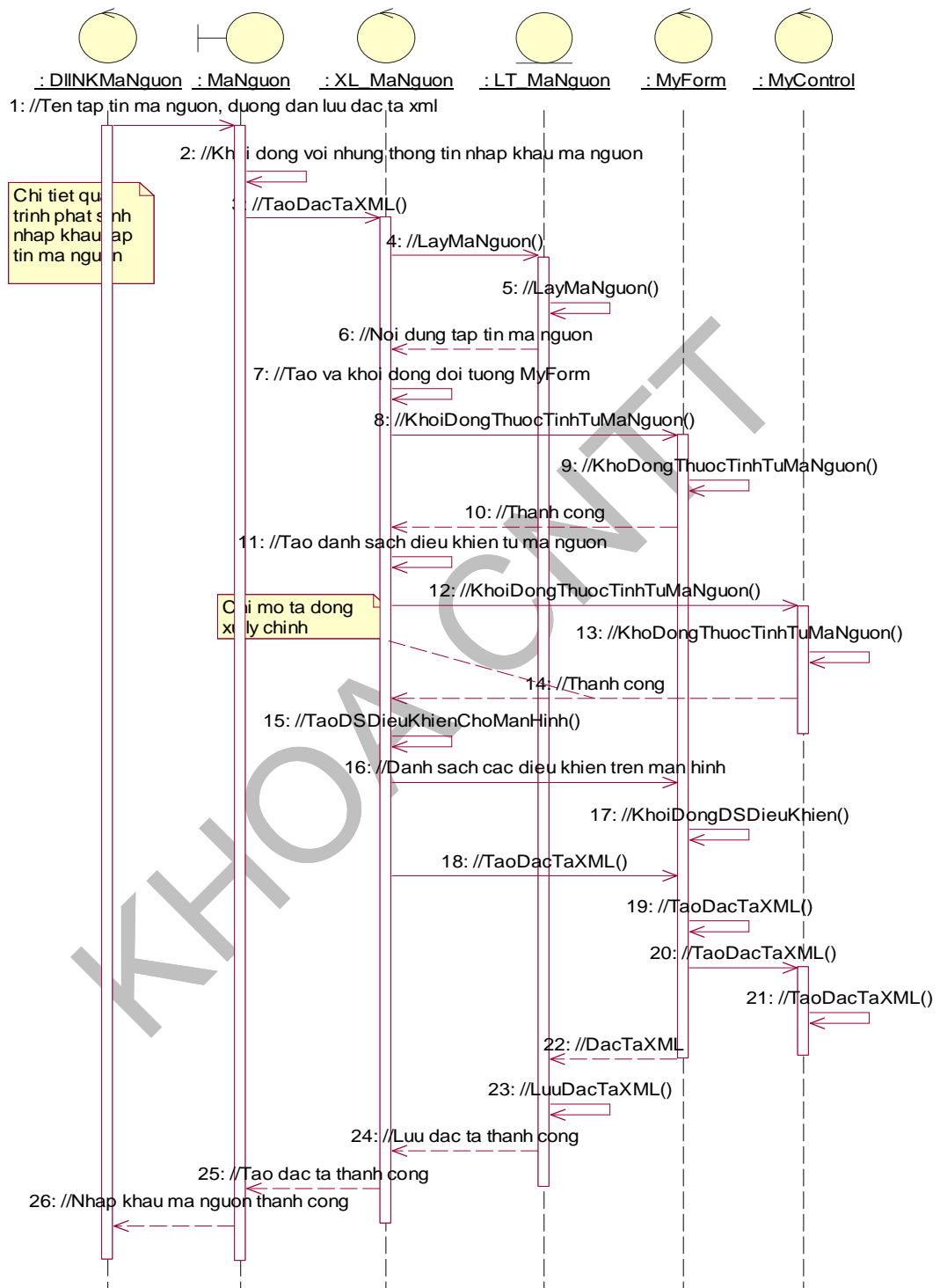
Chức năng này cho phép nhập trực tiếp mã nguồn của một mẫu trên một ngôn ngữ lập trình cụ thể vào hệ thống. Cách thức xử lý như sau:

- Hệ thống chính cho phép người dùng nhập đường dẫn tập tin mã nguồn
- Hệ thống chính xác định module thích hợp sẽ đảm nhận việc nhập mã nguồn này.
- Module thích hợp tiến hành xử lý và trả về cho hệ thống tập tin đặc tả xml.
- Hệ thống chuyển đổi tập tin xml thành giao diện trên công cụ đồ họa ở màn hình MH_QLMau (màn hình quản lý mẫu).



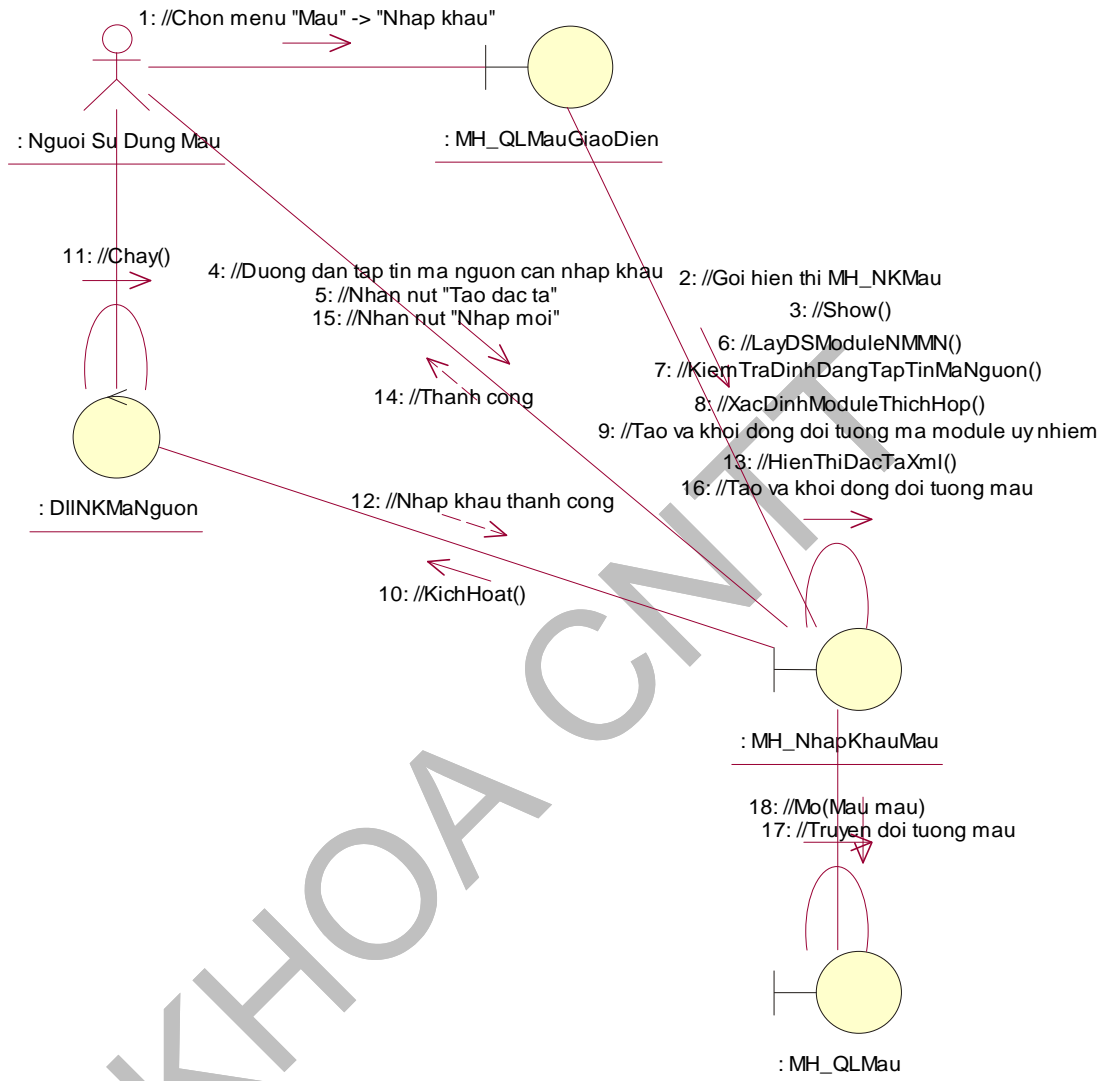
Hình 3-29 Sơ đồ tuần tự xử lý nhập mã nguồn cho mẫu

Dưới đây là chi tiết xử lý nhập mã nguồn được tiến hành trên module nhập mã nguồn.



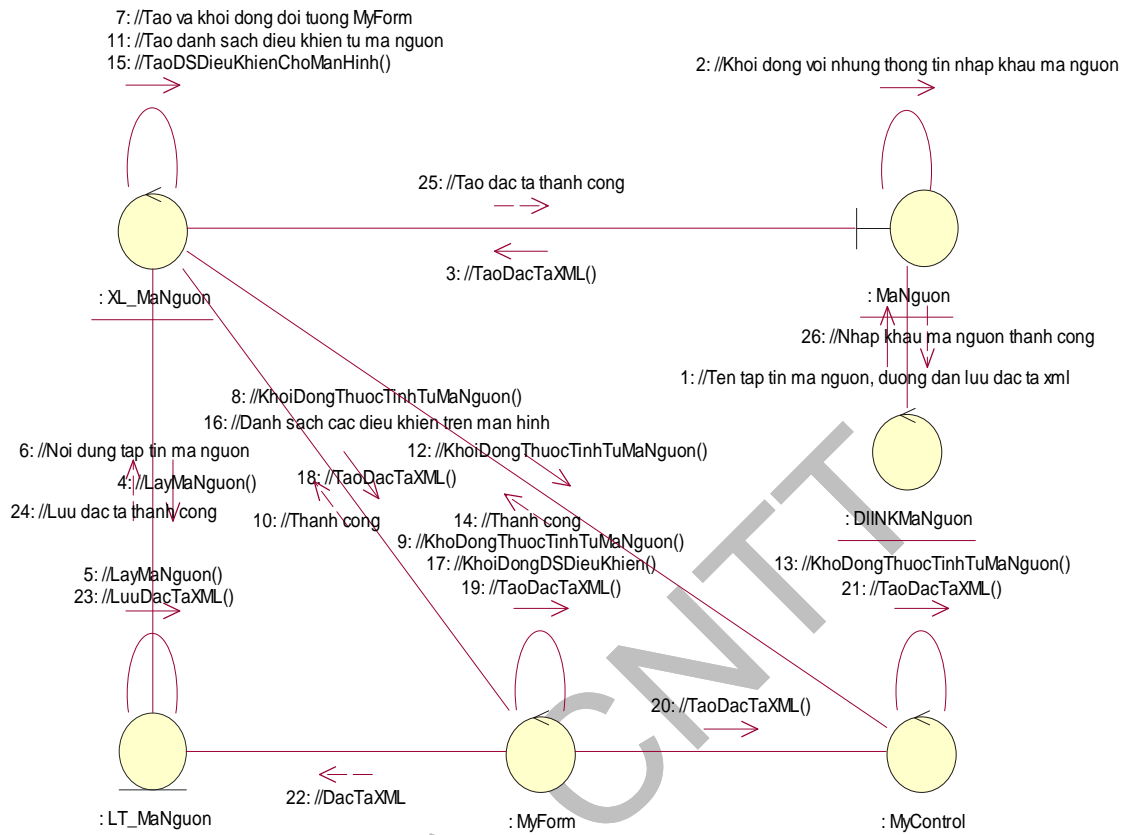
Hình 3-30 Sơ đồ tuần tự chi tiết xử lý nhập mã nguồn trong module

Tương ứng với 2 sơ đồ tuần tự ở trên là 2 sơ đồ tuần tự mô tả xử lý nhập mã nguồn cho Mẫu.



Hình 3-31 Sơ đồ cộng tác xử lý nhập mã nguồn cho Mẫu

Đây là sơ đồ cộng tác mô tả chi tiết nhập mã nguồn trên module.



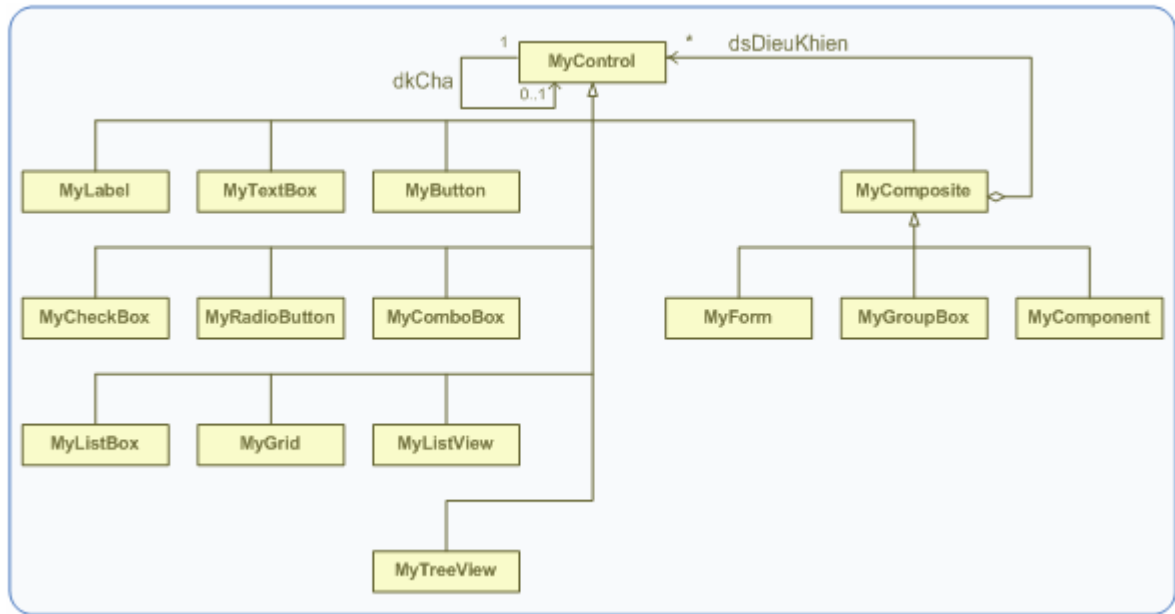
Hình 3-32 Sơ đồ cộng tác chi tiết xử lý nhập mã nguồn

3.2.3 Thiết kế lớp

3.2.3.1 Nhóm các lớp tương ứng với các đối tượng là Điều khiển

- Để tái sử dụng dễ dàng, ta sẽ dùng một lớp trừu tượng MyControl để định nghĩa các thuộc tính chung và các hành vi của nhóm lớp.

- Tương ứng với các điều khiển chuẩn là Label, Button, TextBox, CheckBox, RadioButton, GroupBox, Grid, ListBox, ComboBox, ListView và TreeView; ta có các lớp MyLabel, MyButton, MyTextBox, MyCheckBox, MyRadioButton, MyGroupBox, MyGrid, MyListBox, MyComboBox, MyListView và MyTreeView.



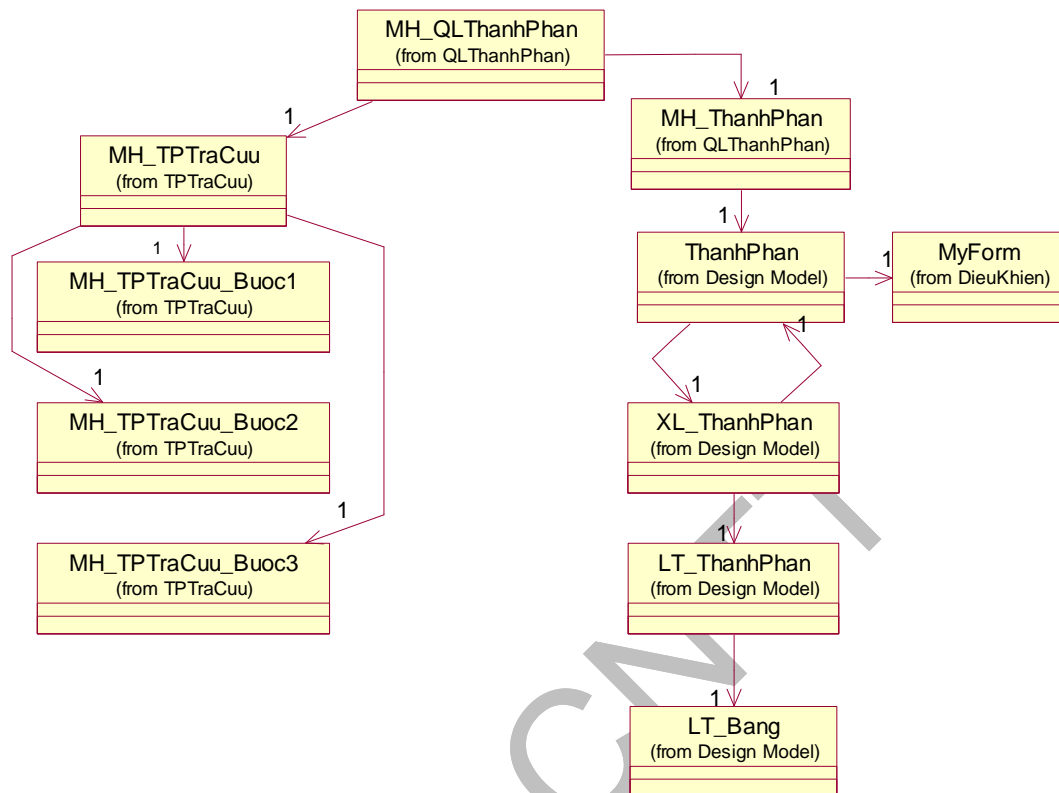
Hình 3-33 Sơ đồ lớp của nhóm các điều khiển

Do các điều khiển có cấu trúc phân cấp theo mối quan hệ tổng thể (MyComposite) và thành phần (MyControl), ta sẽ áp dụng mẫu thiết kế Composit cho nhóm lớp này.

STT	Tên	Diễn giải	Ghi chú
1	MyControl	Lớp MyControl	
2	MyLabel	Lớp MyLabel	
3	MyTextBox	Lớp MyTextBox	
4	MyButton	Lớp MyButton	
5	MyCheckBox	Lớp MyCheckBox	
6	MyRadioButton	Lớp MyRadioButton	
7	MyComboBox	Lớp My ComboBox	
8	MyListBox	Lớp MyListBox	
9	MyGrid	Lớp MyGrid	
10	MyListView	Lớp MyListView	
11	MyTreeView	Lớp MyTreeView	
12	MyComposite	Lớp MyComposite	
13	MyComponent	Lớp MyComponent	
14	MyForm	Lớp MyForm	
15	MyGroupBox	Lớp MyGroupBox	

Bảng 3-10 Danh sách lớp phục vụ chức năng quản lý

3.2.3.2 Thiết kế lớp phục vụ chức năng quản lý thành phần

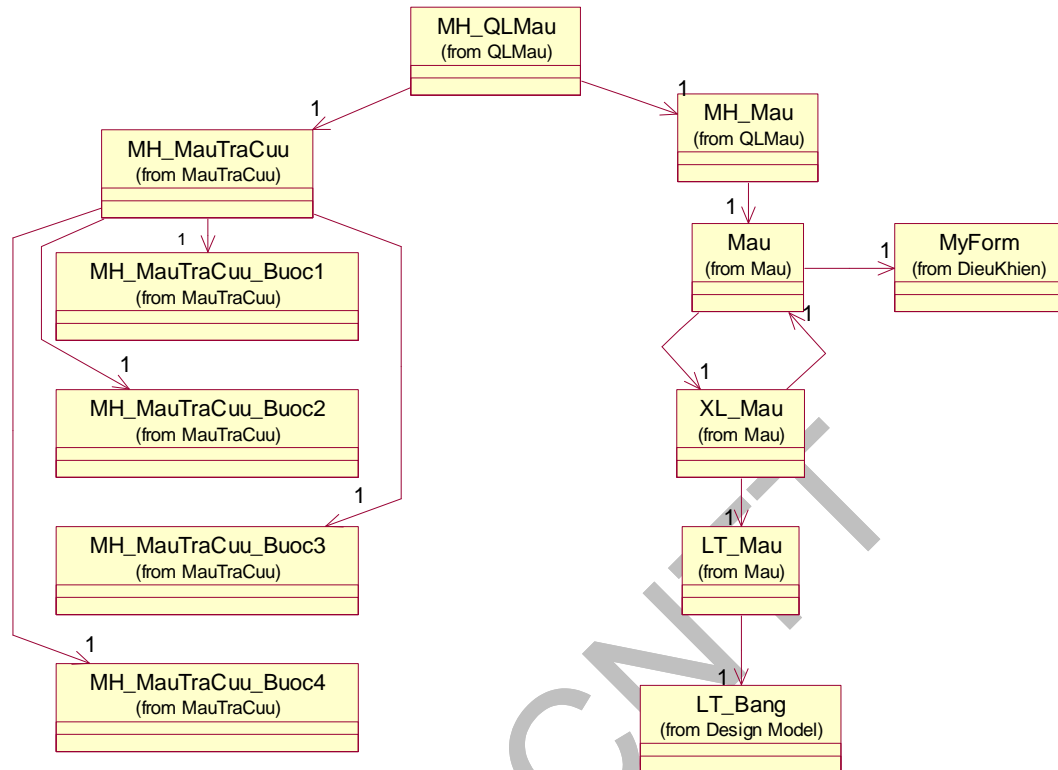


Hình 3-34 Sơ đồ lớp phục vụ chức năng quản lý thành phần

STT	Tên	Diễn giải	Ghi chú
1	MH_QLThanhPhan	Màn hình quản lý thành phần	
2	MH_TPTraCuu	Wizard tra cứu thành phần	
3	MH_TPTraCuu_Buoc1	Màn hình Bước 1 của Wizard tra cứu	
3	MH_TPTraCuu_Buoc2	Màn hình Bước 2 của Wizard tra cứu	
3	MH_TPTraCuu_Buoc3	Màn hình Bước 3 của Wizard tra cứu	
4	ThanhPhan	Lớp ThanhPhan	
5	XL_ThanhPhan	Lớp xử lý của ThanhPhan	
6	LT_ThanhPhan	Lớp lưu trữ của ThanhPhan	
7	LT_Bang	Lớp lưu trữ chung cho các Bảng quan hệ dưới cơ sở dữ liệu	

Bảng 3-11 Danh sách lớp phục vụ chức năng quản lý thành phần

3.2.3.3 Thiết kế lớp phục vụ chức năng quản lý mẫu

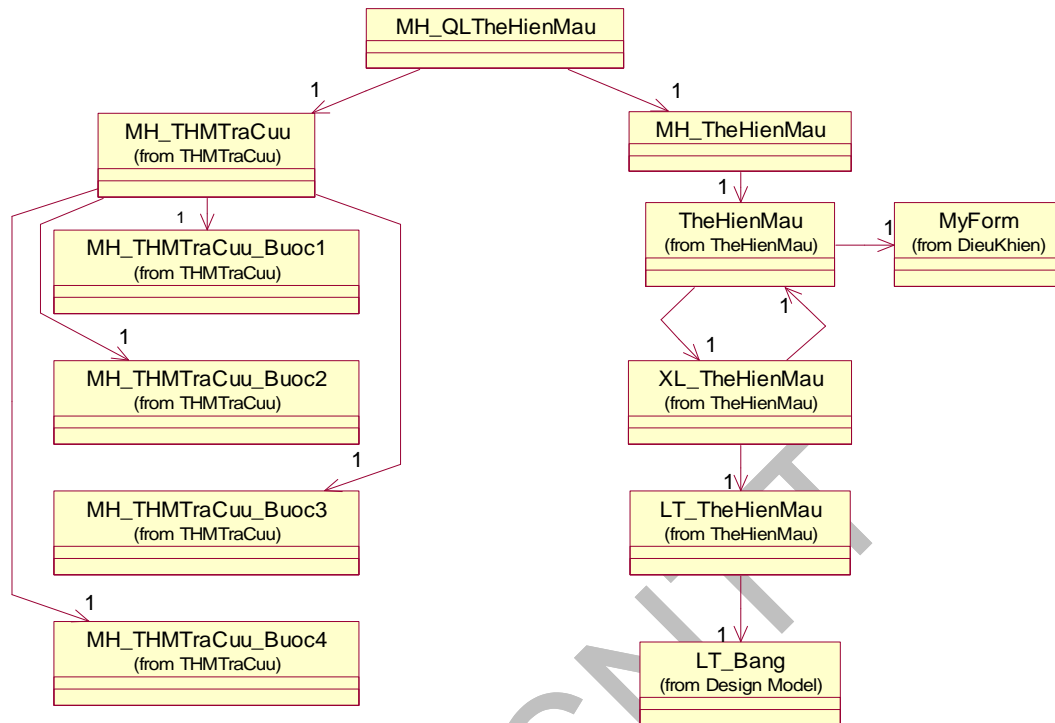


Hình 3-35 Sơ đồ lớp phục vụ chức năng quản lý mẫu

STT	Tên	Diễn giải	Ghi chú
1	MH_QLMau	Màn hình quản lý mẫu	
2	MH_MauTraCuu	Wizard tra cứu mẫu	
3	MH_MauTraCuu_Buoc1	Màn hình Bước 1 của Wizard tra cứu	
4	MH_MauTraCuu_Buoc2	Màn hình Bước 2 của Wizard tra cứu	
5	MH_MauTraCuu_Buoc3	Màn hình Bước 3 của Wizard tra cứu	
6	MH_MauTraCuu_Buoc4	Màn hình Bước 4 của Wizard tra cứu	
7	MH_Mau	Màn hình mẫu	
8	Mau	Lớp Mau	
9	XL_Mau	Lớp xử lý của Mau	
10	LT_Mau	Lớp lưu trữ của Mau	
11	LT_Bang	Lớp lưu trữ chung cho các Bảng quan hệ	
12	MyForm	Lớp MyForm đại diện cho màn hình đồ họa	

Bảng 3-12 Danh sách lớp phục vụ chức năng quản lý mẫu

3.2.3.4 Thiết kế lớp phục vụ chức năng quản lý thể hiện mẫu

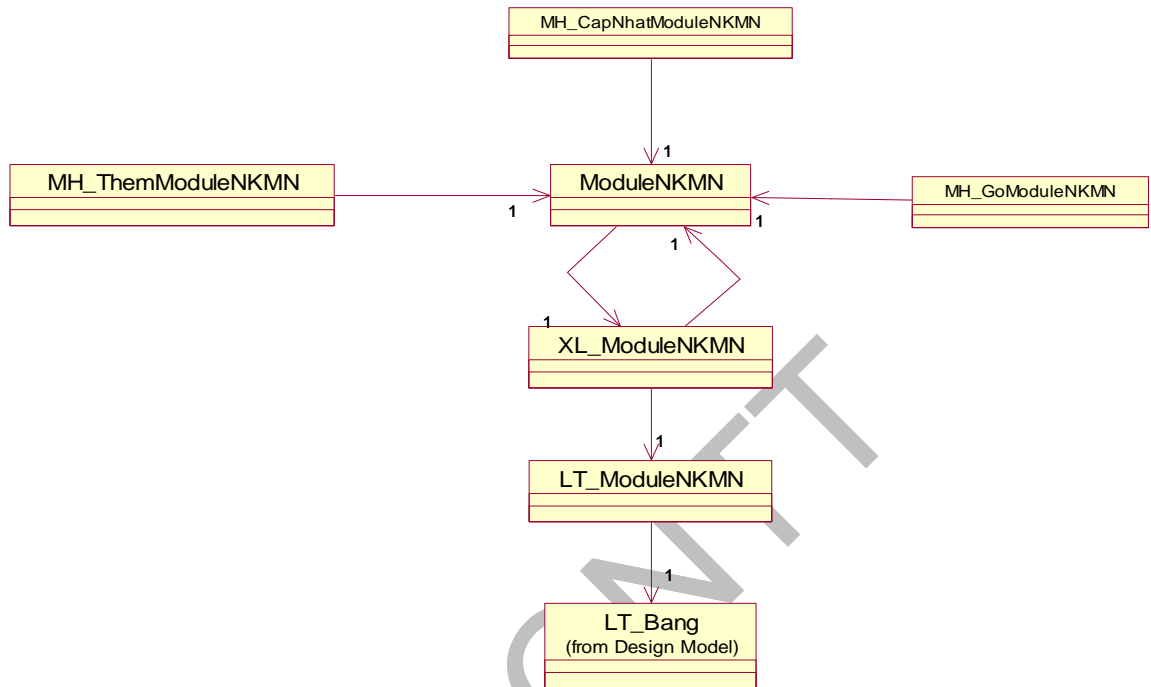


Hình 3-36 Sơ đồ lớp phục vụ chức năng quản lý thành phần

STT	Tên	Diễn giải	Ghi chú
1	MH_QLTheHienMau	Màn hình quản lý thể hiện mẫu	
2	MH_THMTraCuu	Wizard tra cứu thể hiện mẫu	
3	MH_THMTraCuu_Buoc1	Màn hình Bước 1 của Wizard tra cứu	
4	MH_THMTraCuu_Buoc2	Màn hình Bước 2 của Wizard tra cứu	
5	MH_THMTraCuu_Buoc3	Màn hình Bước 3 của Wizard tra cứu	
6	MH_THMTraCuu_Buoc4	Màn hình Bước 4 của Wizard tra cứu	
7	MH_TheHienMau	Màn hình thể hiện mẫu	
8	TheHienMau	Lớp TheHienMau	
9	XL_TheHienMau	Lớp xử lý của TheHienMau	
10	LT_TheHienMau	Lớp lưu trữ của TheHienMau	
11	LT_Bang	Lớp lưu trữ chung cho các Bảng quan hệ	
12	MyForm	Lớp MyForm đại diện cho màn hình đồ họa	

Bảng 3-13 Danh sách lớp phục vụ chức năng quản lý thành phần

3.2.3.5 Thiết kế lớp phục vụ chức năng quản lý module nhập vào mã nguồn

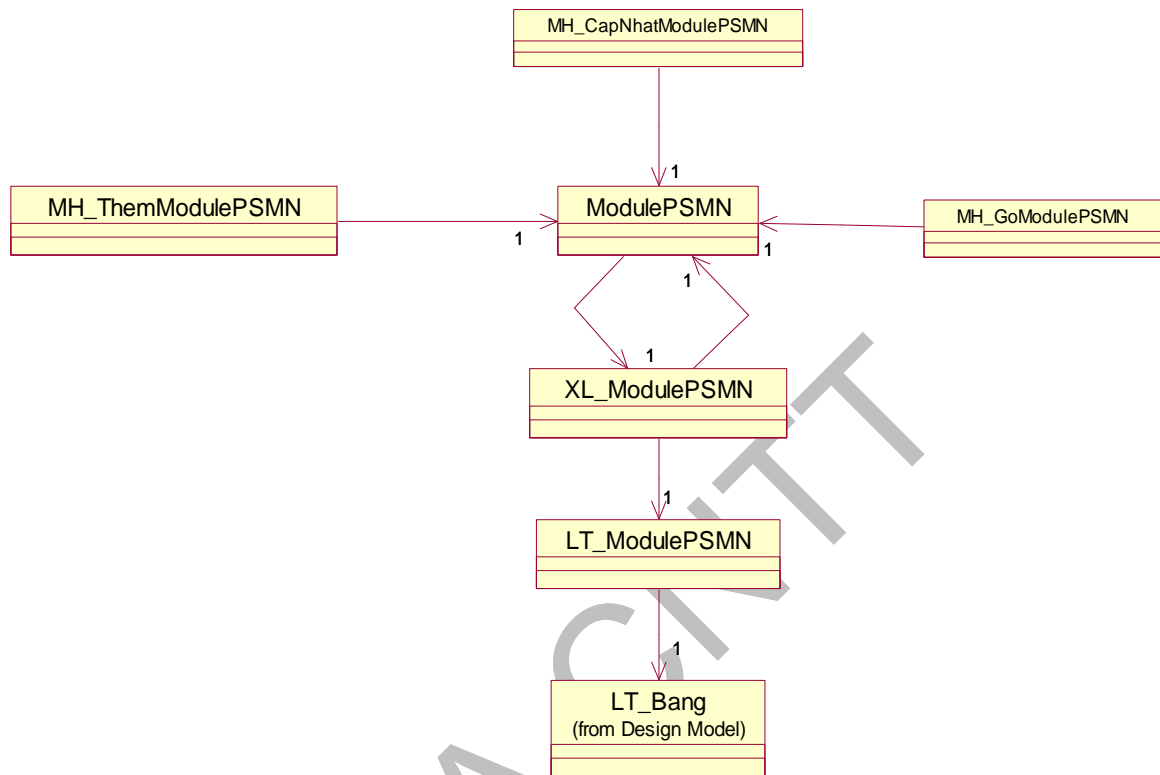


Hình 3-37 Sơ đồ lớp phục vụ chức năng quản lý module nhập vào mã nguồn

STT	Tên	Diễn giải	Ghi chú
1	MH_ThemModuleNKMN	Màn hình thêm module nhập vào mã nguồn	
2	MH_CapNhatModulNKMN	Màn hình cập nhật module nhập vào mã nguồn	
3	MH_GoModuleNKMN	Màn hình gỡ module nhập khẩu mã nguồn	
4	ModuleNKMN	Module nhập vào mã nguồn	
5	XL_ModuleNKMN	Lớp Xử lý module nhập vào mã nguồn	
6	LT_ModuleNKMN	Lớp Lưu trữ module nhập khẩu mã nguồn	
7	LT_Bang	Lớp lưu trữ chung cho các Bảng quan hệ	
8	MH_TheHienMau	Màn hình thể hiện mẫu	

Bảng 3-14 Danh sách lớp phục vụ chức năng quản lý module nhập vào mã nguồn

3.2.3.6 Thiết kế lớp phục vụ chức năng quản lý module phát sinh mã nguồn

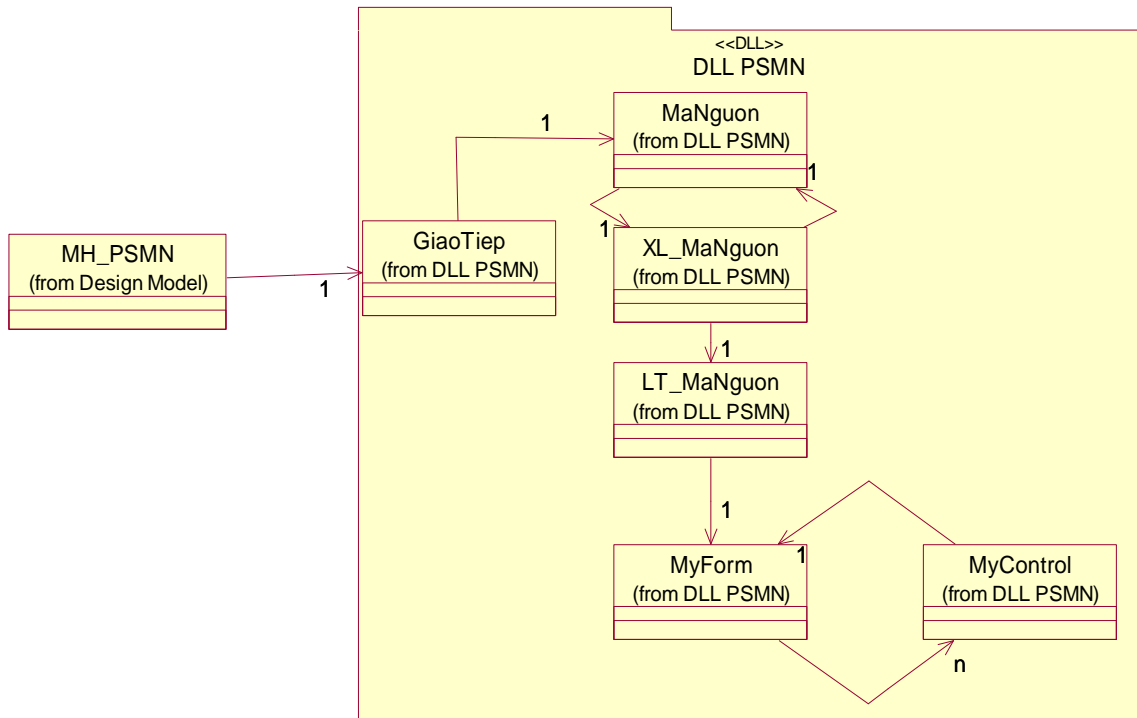


Hình 3-38 Sơ đồ lớp phục vụ chức năng quản lý module phát sinh mã nguồn

STT	Tên	Diễn giải	Ghi chú
1	MH_ThemModulePSMN	Màn hình thêm module phát sinh mã nguồn	
2	MH_CapNhatModulPSMN	Màn hình cập nhật module phát sinh mã nguồn	
3	MH_GoModulePSMN	Màn hình gỡ module nhập khẩu mã nguồn	
4	ModulePSMN	Module phát sinh mã nguồn	
5	XL_ModulePSMN	Lớp Xử lý module phát sinh mã nguồn	
6	LT_ModulePSMN	Lớp Lưu trữ module phát sinh mã nguồn	
7	LT_Bang	Lớp lưu trữ chung cho các Bảng quan hệ	
8	MH_TheHienMau	Màn hình thể hiện mẫu	

Bảng 3-15 Danh sách lớp phục vụ chức năng quản lý module phát sinh mã nguồn

3.2.3.7 Thiết kế lớp phục vụ chức năng phát sinh mã nguồn

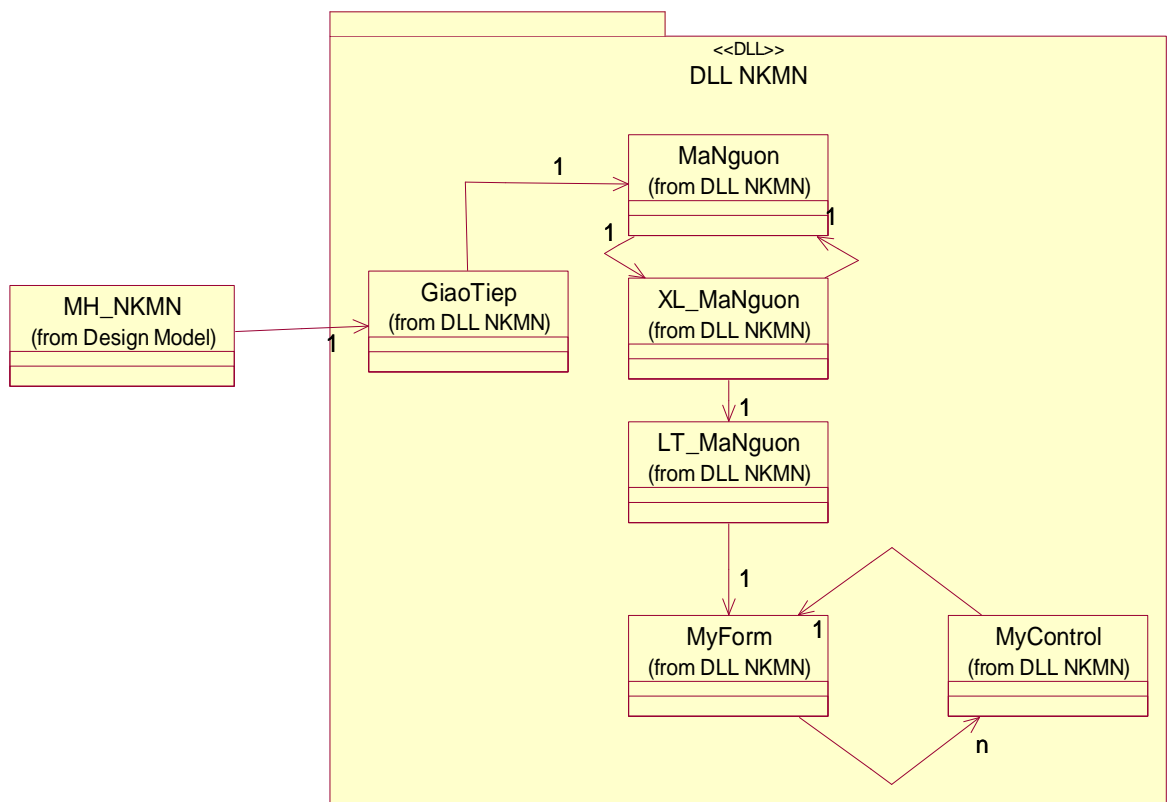


Hình 3-39 Sơ đồ lớp phục vụ chức năng phát sinh mã nguồn

STT	Tên	Diễn giải	Ghi chú
1	MH_PSMN	Màn hình phát sinh mã nguồn	
2	GiaoTiep	Lớp giao tiếp giữa module phát sinh mã nguồn với hệ thống chính	
3	MaNgon	Lớp mã nguồn	
4	XL_MaNgon	Lớp xử lý mã nguồn	
5	LT_MaNgon	Lớp lưu trữ mã nguồn	
6	MyForm	Lớp màn hình đồ họa	
7	MyControl	Lớp MyControl đại diện các điều khiển	

Bảng 3-16 Danh sách lớp phục vụ chức năng phát sinh mã nguồn

3.2.3.8 Thiết kế lớp phục vụ chức năng nhập vào mã nguồn



Hình 3-40 Sơ đồ lớp phục vụ chức năng nhập vào mã nguồn

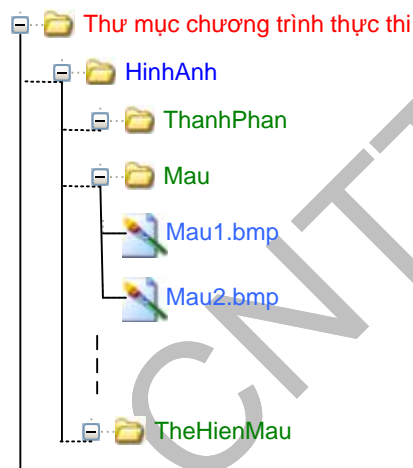
STT	Tên	Diễn giải	Ghi chú
1	MH_NKMN	Màn hình nhập vào mã nguồn	
2	GiaoTiep	Lớp giao tiếp giữa module Nhập vào mã nguồn với hệ thống chính	
3	MaNgon	Lớp mã nguồn	
4	XL_MaNgon	Lớp xử lý mã nguồn	
5	LT_MaNgon	Lớp lưu trữ mã nguồn	
6	MyForm	Lớp màn hình đồ họa	
7	MyControl	Lớp MyControl đại diện các điều khiển	

Bảng 3-17 Danh sách lớp phục vụ chức năng phát sinh mã nguồn

3.2.4 Thiết kế Lưu trữ:

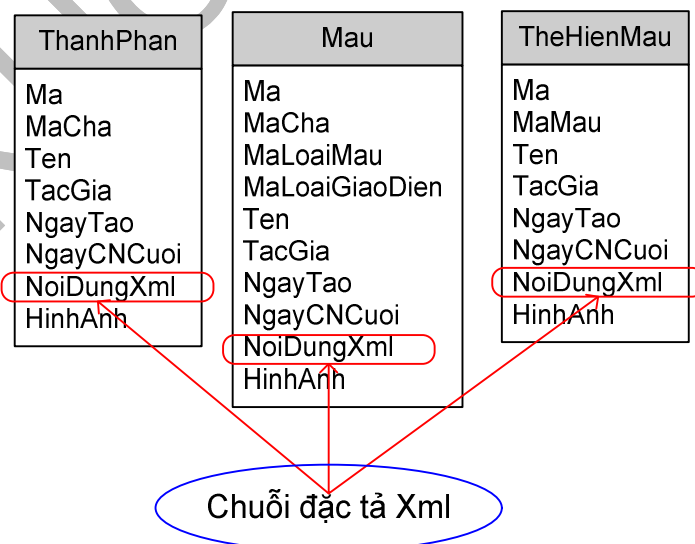
3.2.4.1 Hình thức lưu trữ:

- Dùng Hệ quản trị Cơ sở dữ liệu SQL Server 2000 để quản lý cơ sở dữ liệu với tên “QLMauGiaoDien”.
- Kết hợp thêm việc lưu trữ hình ảnh minh họa cho các thành phần, mẫu, thể hiện mẫu trên tập tin. Tổ chức lưu trữ trên tập tin như sau:



Hình 3-41 Cấu trúc tổ chức thư mục và tập tin lưu hình ảnh minh họa

- Kết hợp việc sử dụng Xml để lưu trữ, cụ thể là các bảng sau có trường NoiDungXml là một chuỗi Xml nhằm đặc tả nội dung của bảng.

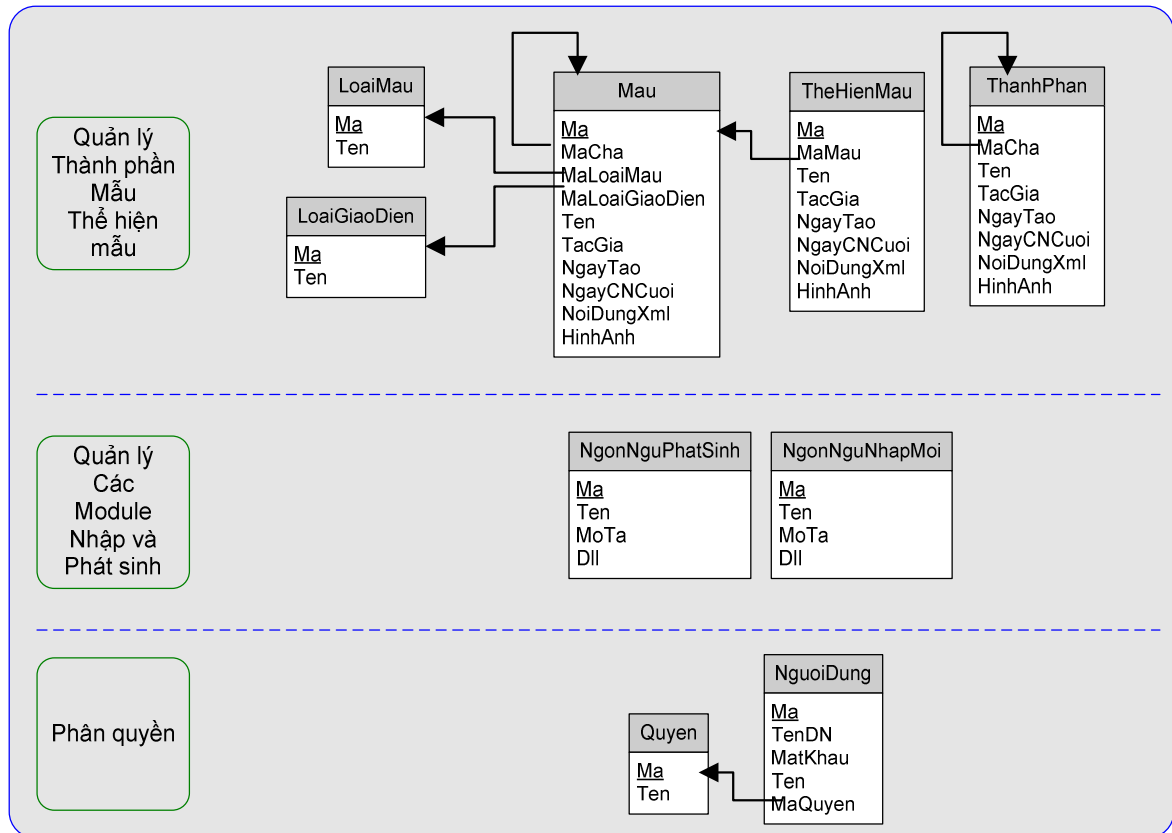


Hình 3-42 Sử dụng Xml trong cơ sở dữ liệu QLMauGiaoDien

3.2.4.2 Vị trí lưu trữ:

- Tập trung trên một máy tính.

3.2.4.3 Sơ đồ Logic:



Hình 3-43 Sơ đồ logic

STT	Tên	Ý nghĩa	Ghi chú
1	Quyen	Quyền có thể có trên hệ thống.	
2	NguoIDung	Người sử dụng chương trình	
3	LoaiMau	Loại mẫu thiết kế giao diện	
4	LoaiGiaoDien	Loại giao diện	
5	ThanhPhan	Thành phần của một mẫu giao diện	
6	Mau	Mẫu giao diện của một loại mẫu cụ thể	
7	TheHienMau	Thể hiện mẫu cụ thể hóa của một màn hình giao diện	
8	NgonNguNhapMoi	Danh sách các ngôn ngữ mà hệ thống hỗ trợ nhập khẩu tập tin mã nguồn	
9	NgonNguPhatSinh	Danh sách ngôn ngữ mà hệ thống hỗ trợ phát sinh mã nguồn	

Bảng 3-18 Danh sách các bảng quan hệ trong cơ sở dữ liệu QLMauGiaoDien

Để biết thêm chi tiết các bảng, nội dung của việc thiết kế chỉ mục và thiết kế store procedure, xin xem thêm trong phục lục.

Chương 4 Cài đặt và thử nghiệm

Nội dung

Chương này sẽ trình bày những chức năng đã được cài đặt và thử nghiệm trên Hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng Hệ thống thông tin.

KHOA CNTT

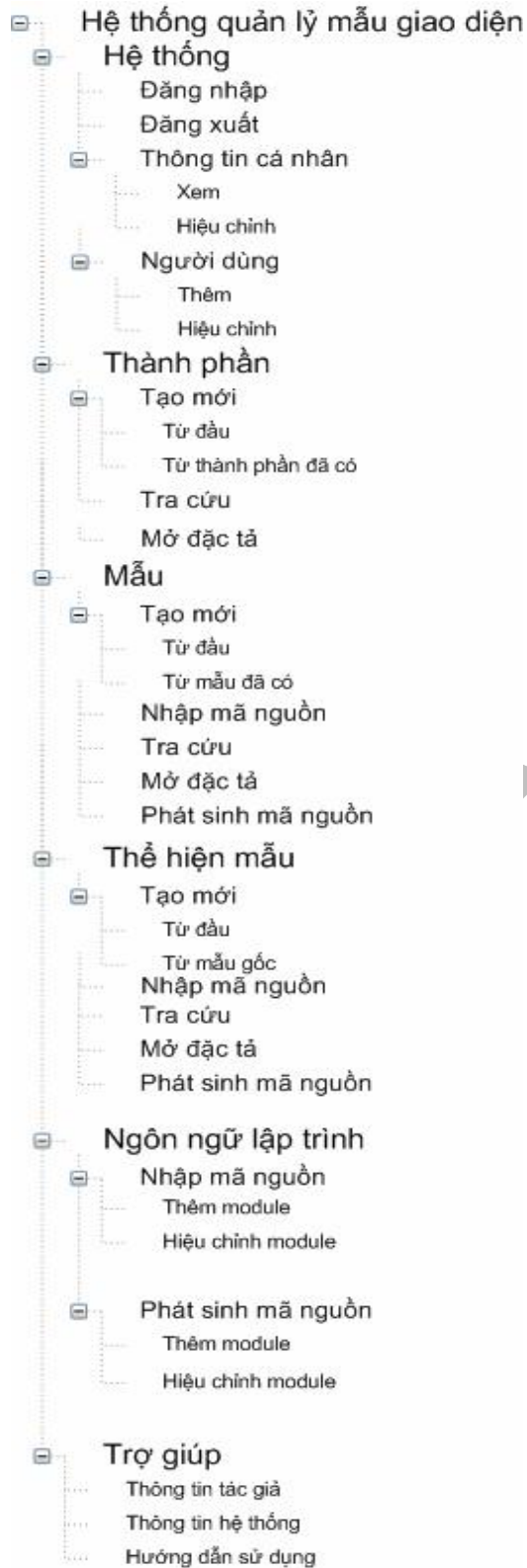
4.1 Môi trường cài đặt

- Hệ điều hành: Windows XP Professional Version 2002 Service Pack 2.
- Môi trường lập trình: Microsoft Visual Studio .Net.
- Ngôn ngữ lập trình: C#.
- Hệ quản trị cơ sở dữ liệu : Microsoft SQL Server 2000 kết hợp với việc sử dụng Xml cho các đối tượng lưu trữ đặc biệt.
- Các thư viện hỗ trợ: Xceed.SmartUI.dll, Xceed.SmartUI.Controls.dll, Xceed.SmartUI.UIStyle.dll.

KHOA CNTT

4.2 Các chức năng được cài đặt

4.2.1 Cây chức năng



4.2.2 Quản lý hệ thống

4.2.2.1 Thêm người dùng vào hệ thống:



Hình 4-1 Màn hình cho phép thêm một người dùng mới

Mô tả cách sử dụng:

Bước 1: Chọn menu “Hệ thống” à chọn “Người dùng” à “Thêm”

Sau bước này màn hình ở Hình 4-1 hiển thị.

Bước 2: Nhập những thông tin cần thiết và nhấn nút “Thêm” để thông tin người dùng mới được lưu xuống cơ sở dữ liệu.

Ghi chú:

Nếu muốn xóa toàn bộ thông tin vừa nhập trên màn hình thì nhấn nút “Bỏ qua”.

4.2.3 Quản lý thông tin cá nhân

4.2.3.1 Xem thông tin cá nhân:

Thông tin cá nhân

THÔNG TIN CÁ NHÂN

Thông tin

Tên đăng nhập	Admin
Mật khẩu	password
Quyền	Nhà quản trị

Thoát

Hình 4-2 Màn hình xem thông tin cá nhân

4.2.3.2 Cập nhật thông tin cá nhân

MH_CapNhatThongTinCaNhan

HIỆU CHỈNH THÔNG TIN CÁ NHÂN

Thông tin

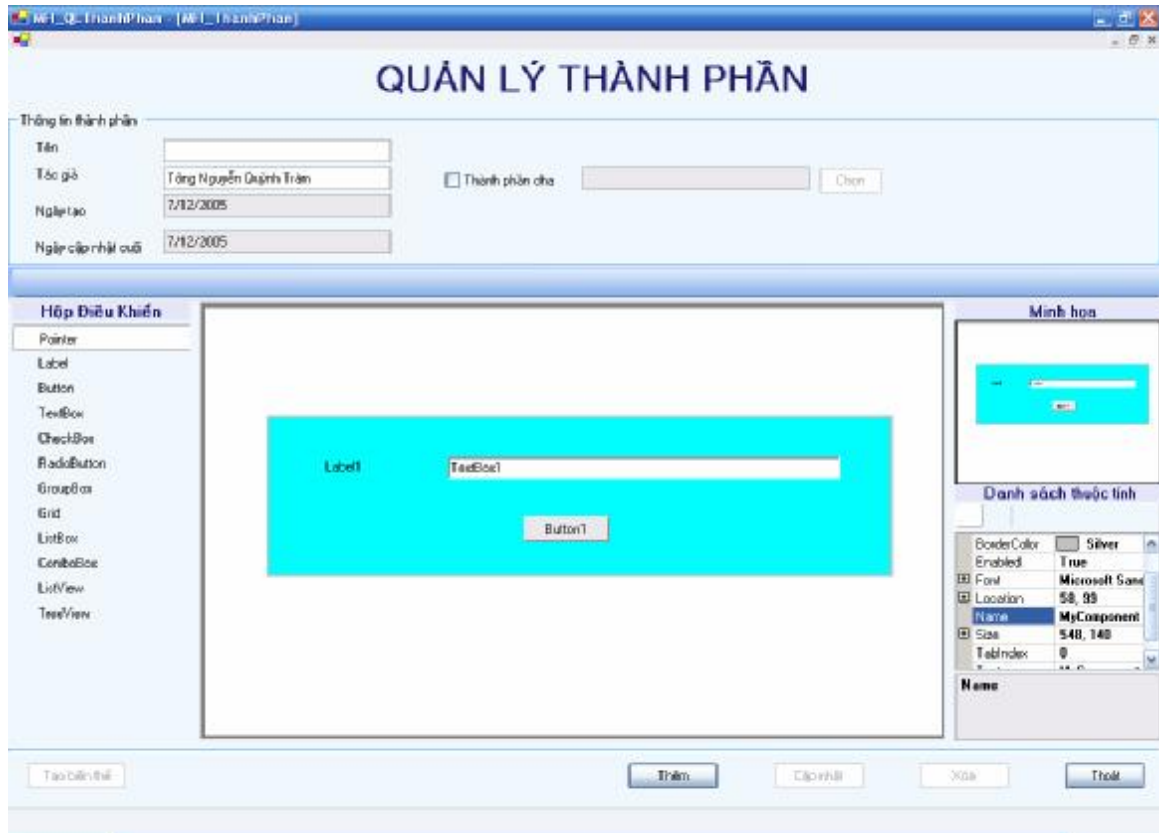
Tên đăng nhập	Phan Văn Vinh
Mật khẩu	password
Quyền	Người sử dụng

Cập nhật Bỏ qua Thoát

Hình 4-3 Màn hình hiệu chỉnh thông tin cá nhân

4.2.4 Quản lý thành phần

4.2.4.1 Thiết kế thành phần bằng công cụ đồ họa trực quan:



Hình 4-4 Màn hình cho phép thiết kế và quản lý thành phần

Mô tả cách sử dụng:

Trên màn hình “Quản lý thành phần” ta có thể thực hiện các chức năng sau:

- Thiết kế thành phần bằng công cụ đồ họa.
- Thêm thành phần vào hệ thống.
- Cập nhật thành phần hiện hành.
- Xóa thành phần hiện hành.
- Tạo biến thể cho thành phần hiện hành.
- Tạo mới từ thành phần cha:

Bước 1: đánh dấu chọn vào checkbox “Thành phần cha”

Bước 2: nhấn nút “Chọn”.

Sau bước này hệ thống sẽ hiển thị wizard để hỗ trợ tra cứu một thành phần sẽ được xem là thành phần cha cho thành phần đang thiết kế.

Bước 3: Tra cứu thành phần theo wizard.

Sau bước này hệ thống sẽ hiển thị thành phần tra cứu được trên màn hình “Quản lý thành phần”.

Bước 4: Thay đổi trên thành phần cha vừa tra cứu được, rồi nhấn nút “Thêm” để yêu cầu hệ thống lưu thành phần vừa tạo.

4.2.4.2 Tra cứu

The screenshot shows a Windows-style application window titled "MH_TPTraCuu". The main area is titled "TRA CỨU THÀNH PHẦN". Below the title, there is a text box with instructions: "Lọc bớt các thành phần bằng cách chọn ra 1 cây T là con của cây 'Thành phần'. Bạn hãy chỉ định gốc của cây T từ cây 'Thành phần'. Qui trình sẽ tiếp tục tra cứu trên cây T".

On the left side, there is a vertical blue bar with three buttons: "Chọn thành phần gốc", "Nhập thông tin", and "Chọn kết quả".

The main content area is divided into two sections. The top section is titled "Thành phần" and contains a tree view with the following structure:

- Thành phần
 - Tên màn hình
 - Nhập liệu
 - BT của tên màn hình
 - Nút chức năng

The bottom section is titled "Thông tin thành phần" and contains several input fields:

- Tên:
- Tác giả:
- Ngày tạo:
- Ngày cập nhật cuối:
- Thành phần cha:

At the bottom right, there is a pink button labeled "TÊN MÀN HÌNH".

At the bottom of the window, there are four buttons: "Trở lại", "Tiếp tục", "Hoàn tất", and "Bỏ qua".

Hình 4-5 Wizard cho phép tra cứu thành phần

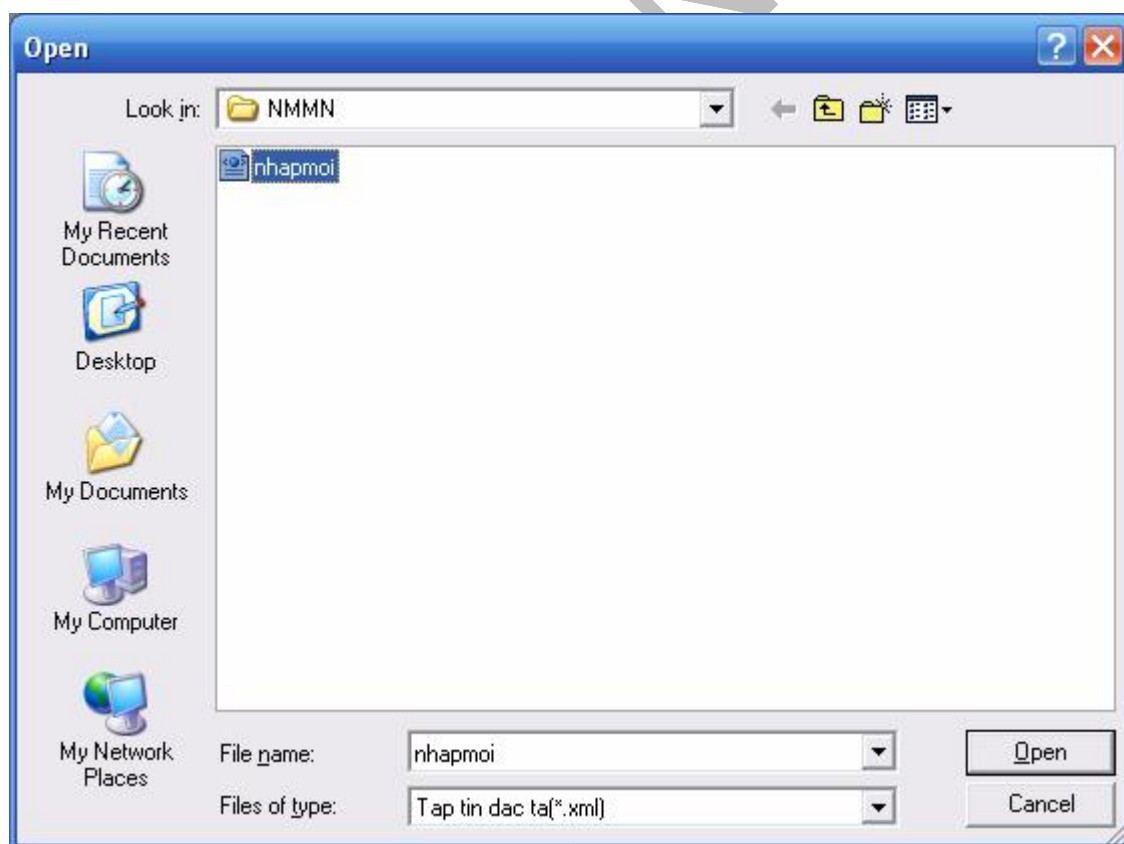
Mô tả cách sử dụng:

Bước 1: Chọn một thành phần trên cây Thành phần. Ứng với thành phần được chọn hệ thống sẽ hiển thị tất cả những thành phần con được tạo ra từ thành phần đang chọn. Chọn một thành phần mà ta muốn tìm kiếm trên những thành phần con của nó. Nhấn nút “Tiếp tục”.

Bước 2: Nhập thông tin mô tả để tìm kiếm và nhấn “Tiếp tục”.

Bước 3: Chọn một thành phần cần tìm trong số danh sách những thành phần kết quả được liệt kê. Mỗi khi một thành phần được chọn, hệ thống hiển thị thông tin chi tiết của thành phần để hỗ trợ chọn lựa dễ dàng và chính xác. Sau đó nhấn nút “Hoàn tất”. Thành phần chọn sẽ được hiển thị trong màn hình “Quản lý Thành phần” như Hình 4-4.

4.2.4.3 Mở đặc tả



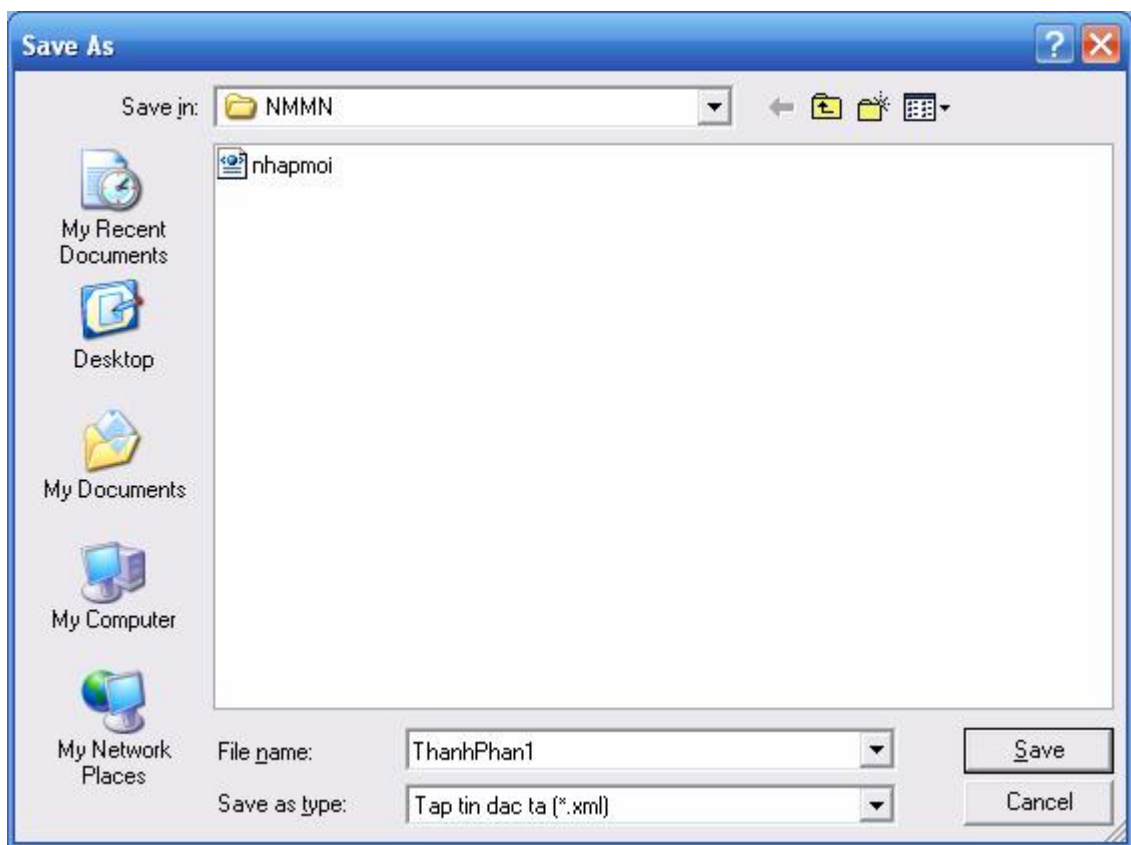
Hình 4-6 Màn hình mở đặc tả xml cho thành phần

Mô tả cách sử dụng:

Bước 1: Chọn menu “Thành phần” à “Mở đặc tả”.

Bước 2: Chọn tập tin .xml cần mở. Nhấn “Open”. Hệ thống sẽ hiển thị thành phần được đặc tả trong tập tin lên màn hình Hình 4-4.

4.2.4.4 Lưu đặc tả



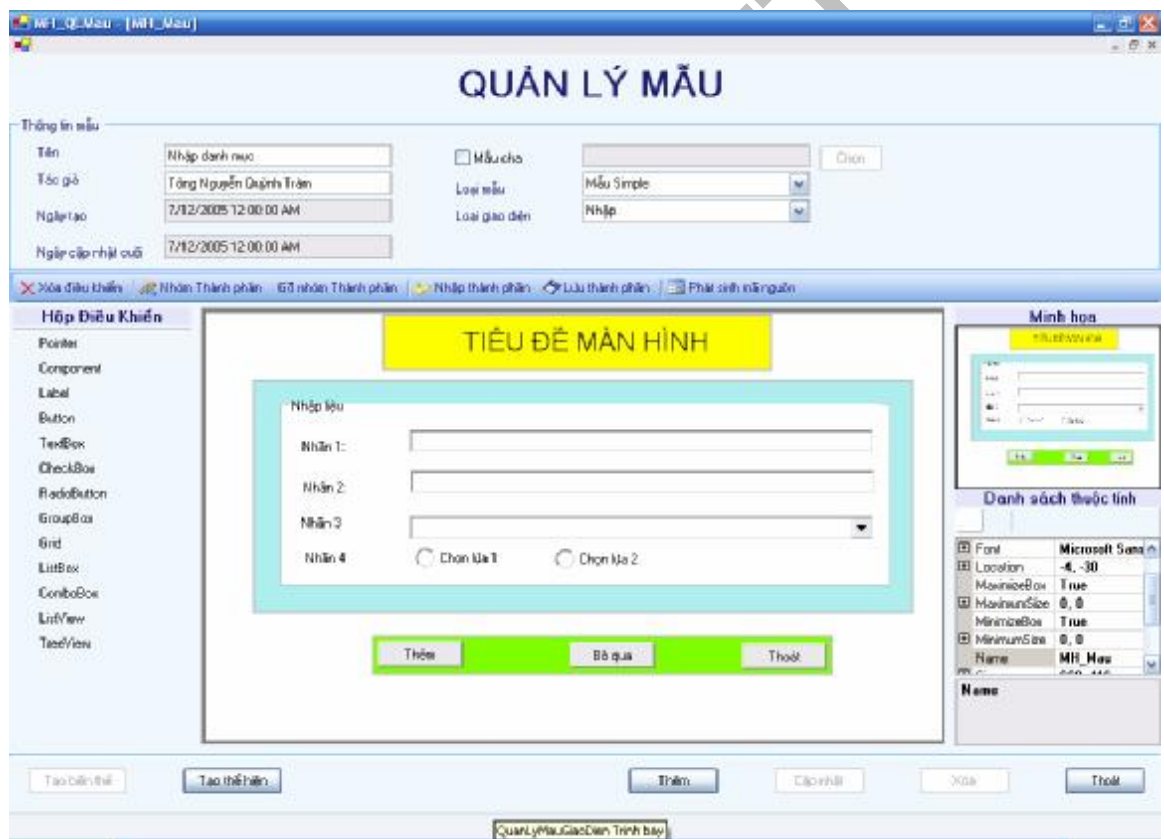
Mô tả cách sử dụng:

Bước 1: Chọn chức năng “Lưu đặc tả” trong màn hình Hình 4-4 cho thành phần đang được thiết kế.

Bước 2: Đặt tên cho tập tin .xml. Nhấn “Save” để lưu thành phần.

4.2.5 Quản lý mẫu

4.2.5.1 Thiết kế mẫu bằng công cụ đồ họa trực quan:



Hình 4-7 Màn hình quản lý mẫu

Mô tả cách sử dụng:

Trên màn hình “Quản lý mẫu” ta có thể thực hiện các chức năng sau:

- Thiết kế mẫu bằng công cụ đồ họa.
- Thêm mẫu vào hệ thống.

- Cập nhật mẫu hiện hành.
- Xóa mẫu hiện hành.
- Tạo biến thể cho mẫu hiện hành.
- Tạo mới từ mẫu gốc.
- Nhóm các điều khiển thành một thành phần.
- Gỡ thành phần thành các điều khiển độc lập.
- Nhập một thành phần vào mẫu.
- Xuất một thành phần của mẫu đang thiết kế để hệ thống lưu trữ.
- Phát sinh mã nguồn cho mẫu hiện hành.

4.2.5.2 Tra cứu mẫu

Chọn một loại mẫu từ đây bằng cách nhập chọn. Sau đó chọn một loại giao diện phù hợp

Loại mẫu giao diện

- Mẫu Simple
- Mẫu Master-Detail

Thông tin loại mẫu

Tên: Mẫu Simple

Thông tin: Các màn hình theo mẫu Simple gồm có 3 thành phần chính: Tiêu đề, Hình ảnh thông tin và Các nút chức năng

Chọn loại giao diện

Loại giao diện: Nhập

Trở lại Tiếp tục Huỷ bỏ Bỏ qua

Hình 4-8 Tra cứu mẫu

Mô tả cách sử dụng:

Bước 1: Chọn menu “Mẫu” à “Tra cứu”.

Bước 2: Chọn một mẫu trên cây tra cứu. Hệ thống sẽ hiển thị tất cả các mẫu được tạo ra từ mẫu đang chọn. Chọn một mẫu trong khung hiển thị để tra cứu trên các mẫu được hình thành từ mẫu đã chọn. Nếu không chọn mẫu trong khung hiển thị thì hệ thống sẽ tra cứu trên những mẫu được hình thành từ mẫu chọn trong cây hiển thị.

Bước 3: Nhập thông tin cần tra cứu. Nếu không nhập thì hệ thống sẽ xem tất cả các mẫu trong phạm vi tìm kiếm là thỏa mãn yêu cầu tìm kiếm.

Bước 4: Chọn một mẫu trong số các mẫu được hiển thị trong khung kết quả, và nhấn nút “Hòan tất”.

Sau bước này, mẫu được chọn sẽ được hiển thị trong màn hình Quản lý mẫu như hình Hình 4-7.

4.2.5.3 Cập nhật mẫu

Mô tả cách sử dụng:

Bước 1: Chọn menu “Mẫu” à “Cập nhật”.

Sau bước này hệ thống sẽ hiển thị wizard tra cứu mẫu như hình Hình 4-8.

Bước 2: Tra cứu để được mẫu cần cập nhật.

Bước 3: Mẫu cần cập nhật được hiển thị trong màn hình Hình 4-7. Thay đổi mẫu và nhấn nút “Cập nhật”.

4.2.5.4 Xóa mẫu

The screenshot shows the 'XÓA MẪU' (Delete Template) interface. The sidebar on the left includes options for selecting template types, choosing a template, entering search information, and selecting results. The main content area is divided into three parts: a template selection area with a 'Mẫu nhập thông tin' box and a 'Màn hình nhập 2' box; a 'Thông tin thể hiện mẫu' (Template Display Information) section with fields for Name, Author, Creation Date, and Last Update Date; and a 'Hình minh họa' (Illustration) section showing a preview of the template form. The preview window is titled 'TÊN MÀN HÌNH' and displays a form with various input fields and buttons. At the bottom of the window are five buttons: 'Xóa', 'Trở lại', 'Tiếp tục', 'Hoàn tất', and 'Bỏ qua'.

Hình 4-9 Màn hình xóa mẫu

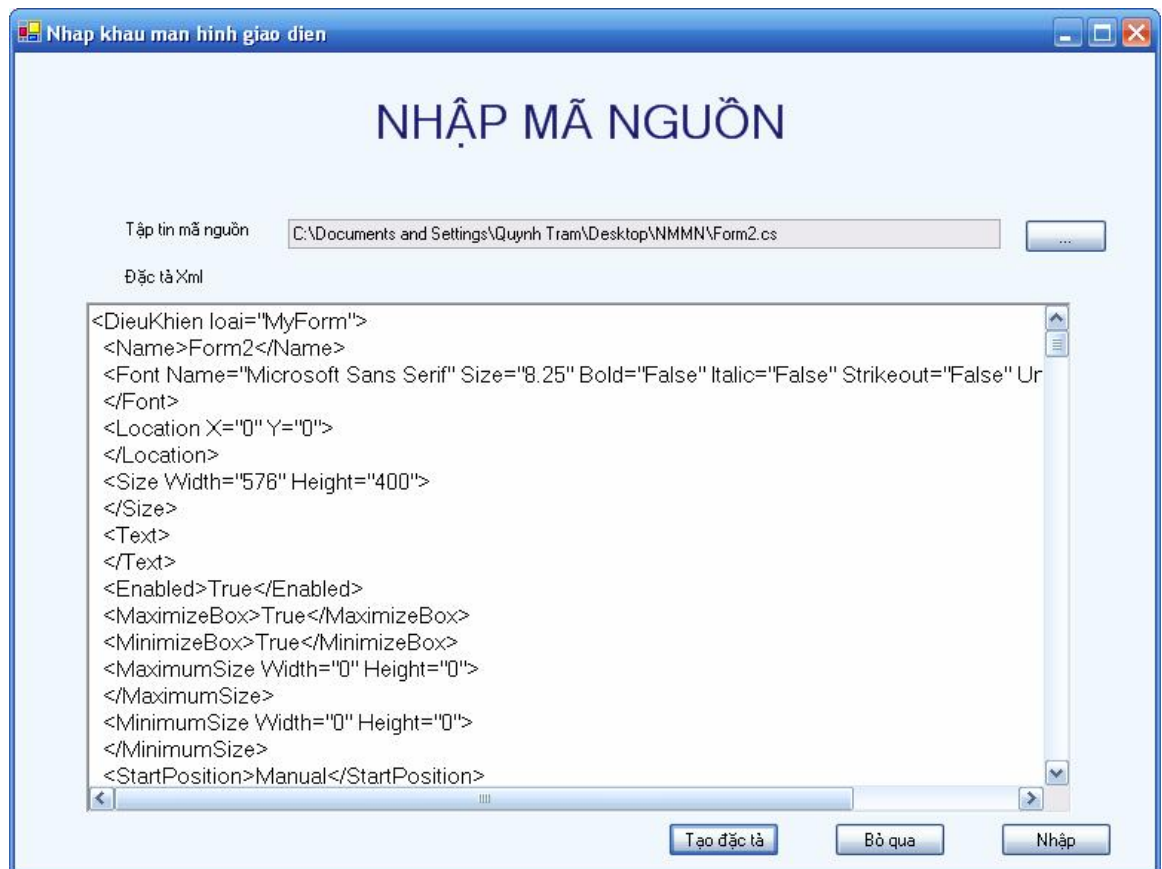
4.2.5.5 Mở đặc tả

Tương tự chức năng mở đặc tả của thành phần mẫu.

4.2.5.6 Lưu đặc t

Tương tự chức năng mở đặc tả của thành phần mẫu.

4.2.5.7 Nhập mã nguồn



Hình 4-10 Màn hình nhập mã nguồn

Mô tả cách sử dụng:

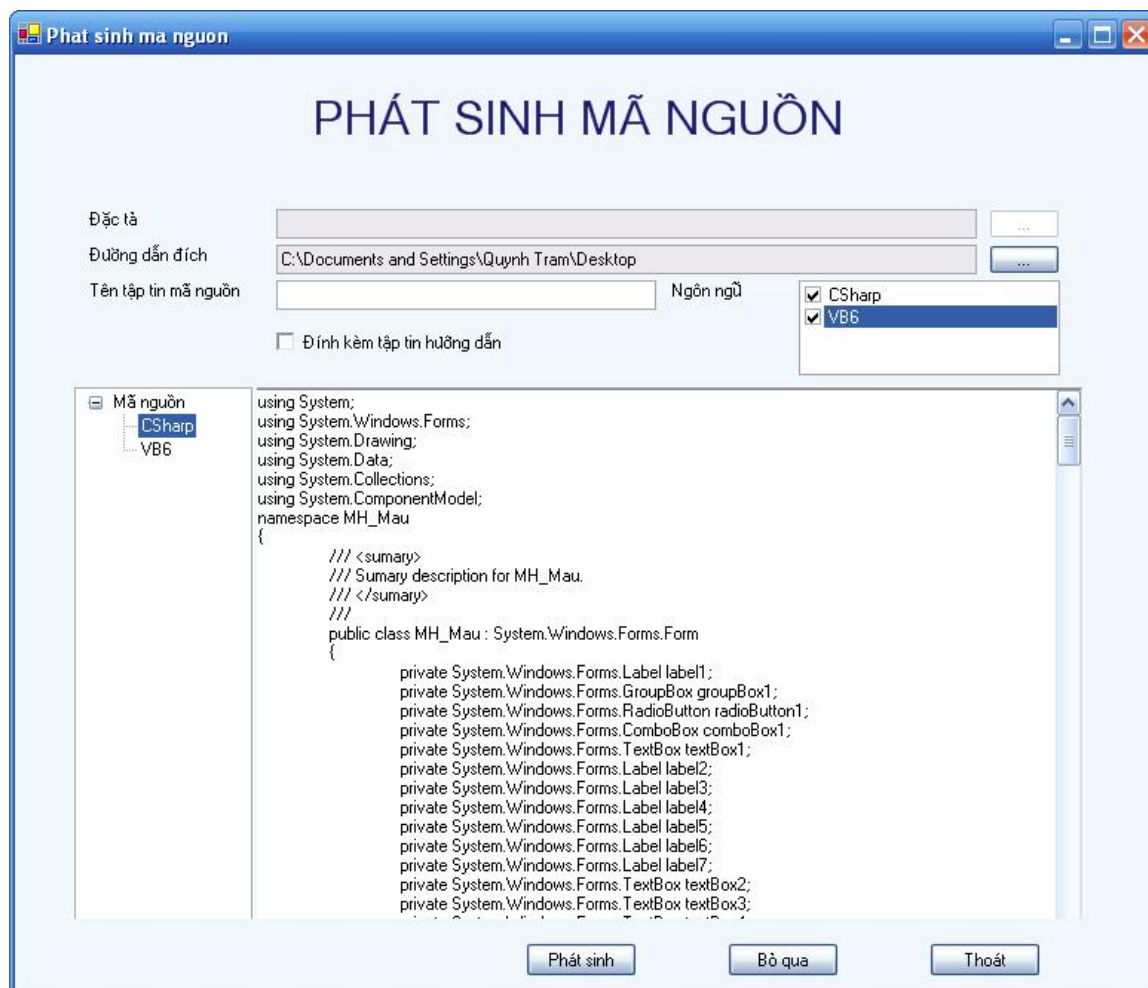
Bước 1: Chọn menu “Mẫu” à “Nhập mã nguồn”.

Bước 2: Chọn đường dẫn cho tập tin mã nguồn.

Bước 3: Nhấn nút “Tạo đặc tả”. Hệ thống sẽ gọi module nhập mã nguồn thích hợp để tạo đặc tả xml cho màn hình giao diện của tập tin mã nguồn, và hiển thị kết quả.

Bước 4: Nhấp nút “Nhập” để hiển thị mẫu vừa nhập từ tập tin mã nguồn vào màn hình Hình 4-7.

4.2.5.8 Phát sinh mã nguồn



Hình 4-11 Màn hình phát sinh mã nguồn

Mô tả cách sử dụng:

Cách 1: Phát sinh mã nguồn cho tập tin đặc tả xml của mẫu.

Bước 1: Chọn menu “Mẫu” à “Phát sinh mã nguồn”.

Bước 2: Chọn đường dẫn đến tập tin mã nguồn.

Bước 3: Nhập tên cho tập tin mã nguồn được phát sinh.

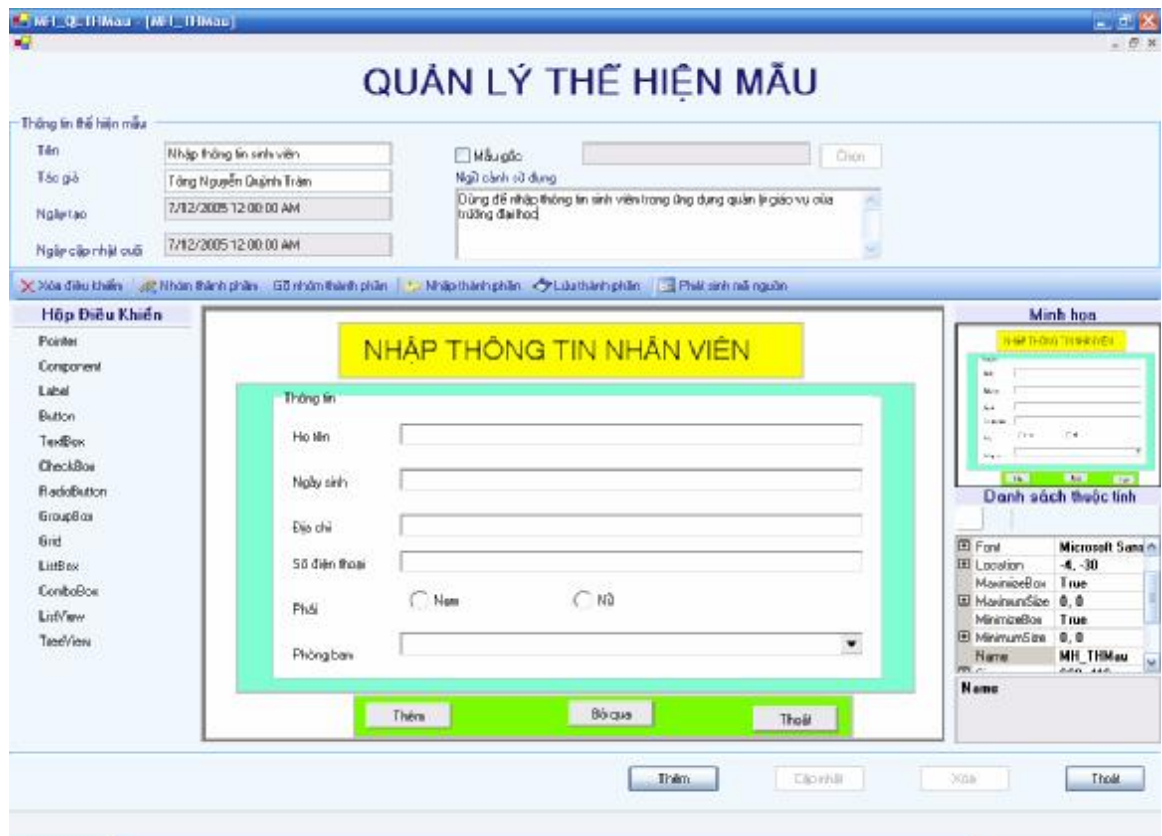
Bước 4: Chọn ngôn ngữ lập trình cần phát sinh. Có thể chọn một lúc nhiều ngôn ngữ lập trình. Có muốn đính kèm tập tin hướng dẫn sử dụng mã nguồn không.

Bước 5: Nhấn nút “Phát sinh”.

Cách 2: Phát sinh mã nguồn cho thể hiện mẫu đang thiết kế. Chọn chức năng “Phát sinh mã nguồn” trong màn hình Hình 4-7.

4.2.6 Quản lý thể hiện mẫu

4.2.6.1 Thiết kế thể hiện mẫu bằng công cụ đồ họa trực quan:



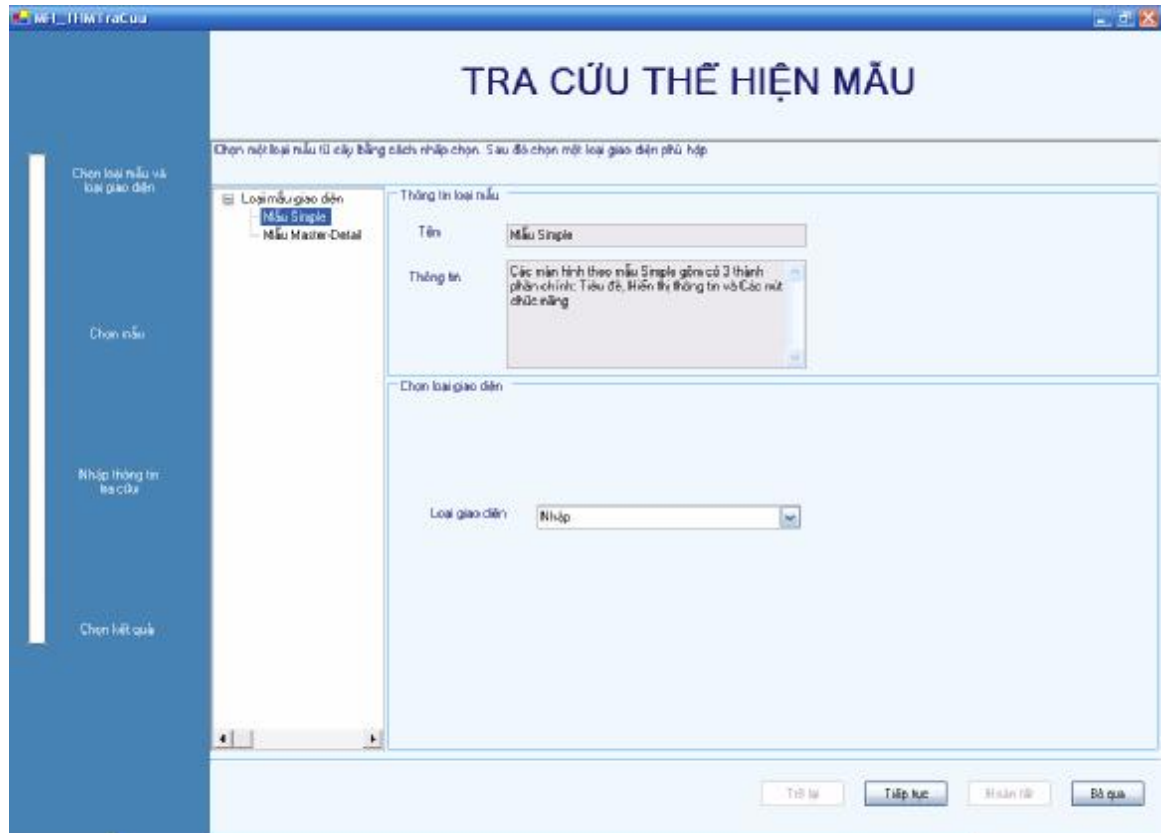
Hình 4-12 Màn hình quản lý Thể hiện mẫu

Trên màn hình “Quản lý thể hiện mẫu” ta có thể thực hiện các chức năng sau:

- Thiết kế thể hiện mẫu bằng công cụ đồ họa.
- Thêm thể hiện mẫu vào hệ thống.
- Cập nhật thể hiện mẫu hiện hành.
- Xóa thể hiện mẫu hiện hành.
- Tạo biến thể cho thể hiện mẫu hiện hành.
- Tạo mới từ mẫu gốc.
- Nhóm các điều khiển thành một thành phần.
- Gỡ thành phần thành các điều khiển độc lập.
- Nhập một thành phần vào thể hiện mẫu.
- Xuất một thành phần của thể hiện mẫu đang thiết kế để hệ thống lưu trữ.

- Phát sinh mã nguồn cho thể hiện mẫu hiện hành.

4.2.6.2 Tra cứu thể hiện mẫu



Hình 4-13 Màn hình tra cứu thể hiện mẫu

Mô tả cách sử dụng:

Bước 1: Chọn menu “Thể hiện mẫu” à “Tra cứu”.

Bước 2: Chọn một mẫu trên cây tra cứu. Hệ thống sẽ hiển thị tất cả các mẫu được tạo ra từ mẫu đang chọn. Chọn một mẫu trong khung hiển thị để tra cứu trên các thể hiện mẫu được hình thành từ mẫu đã chọn. Nếu không chọn mẫu trong khung hiển thị thì hệ thống sẽ tra cứu trên những thể hiện mẫu được hình thành từ mẫu chọn trong cây hiển thị.

Bước 3: Nhập thông tin cần tra cứu. Nếu không nhập thì hệ thống sẽ xem tất cả các thể hiện mẫu trong phạm vi tìm kiếm là thỏa mãn yêu cầu tìm kiếm.

Bước 4: Chọn một thể hiện mẫu trong số các mẫu được hiển thị trong khung kết quả, và nhấn nút “Hoàn tất”.

Sau bước này, mẫu được chọn sẽ được hiển thị trong màn hình Quản lý mẫu như hình Hình 4-12.

4.2.6.3 Cập nhật thể hiện mẫu

Mô tả cách sử dụng:

Bước 1: Chọn menu “Thể hiện mẫu” à “Cập nhật”.

Sau bước này hệ thống sẽ hiển thị wizard tra cứu thể hiện mẫu như hình Hình 4-13.

Bước 2: Tra cứu để được thể hiện mẫu cần cập nhật.

Bước 3: Mẫu cần cập nhật được hiển thị trong màn hình Hình 4-12. Thay đổi mẫu và nhấn nút “Cập nhật”.

4.2.6.4 Xóa thể hiện mẫu:

The screenshot shows a software window titled "MH_THMTraCuu" with a main heading "XÓA THỂ HIỆN MẪU". On the left is a blue sidebar with four menu items: "Chọn loại mẫu và loại giao diện", "Chọn mẫu", "Nhập thông tin tra cứu", and "Chọn kết quả". The main content area is divided into several sections. At the top, it says "Kết quả tra cứu". Below that is a section titled "Thể hiện mẫu" which contains a text box with the value "Nhập tỉnh thành". To the right of this is a section titled "Thông tin thể hiện mẫu" containing several input fields: "Tên" (with value "Nhập tỉnh thành"), "Tác giả" (with value "Tống Nguyễn Quỳnh Trâm"), "Ngày tạo" (with value "7/8/2005"), "Ngày cập nhật cuối", and "Ngữ cảnh sử dụng" (with a dropdown menu). Below this is a section titled "Hình ảnh minh họa" which shows a small thumbnail of a form titled "NHẬP TỈNH THÀNH". At the bottom of the window are five buttons: "Xóa", "Trở lại", "Tiếp tục", "Hoàn tất", and "Bỏ qua".

Hình 4-14 Màn hình xóa thể hiện mẫu

4.2.6.5 Mở đặc tả

Tương tự chức năng mở đặc tả của thành phần mẫu.

4.2.6.6 Lưu đặc tả

Tương tự chức năng mở đặc tả của thành phần mẫu.

4.2.6.7 Phát sinh mã nguồn

Mô tả cách sử dụng:

Cách 1: Phát sinh mã nguồn cho tập tin đặc tả của thể hiện mẫu.

Bước 1: Chọn menu “Thể hiện mẫu” à “Phát sinh mã nguồn”.

Các bước sau tương tự chức năng phát sinh mã nguồn cho tập tin đặc tả của mẫu.

Cách 2: Phát sinh mã nguồn cho thể hiện mẫu đang thiết kế.

Bước 1: Chọn chức năng “Phát sinh mã nguồn” trong hình Hình 4-12.

Các bước sau tiến hành tương tự chức năng phát sinh mã nguồn cho mẫu đang thiết kế.

4.2.6.8 Nhập mã nguồn

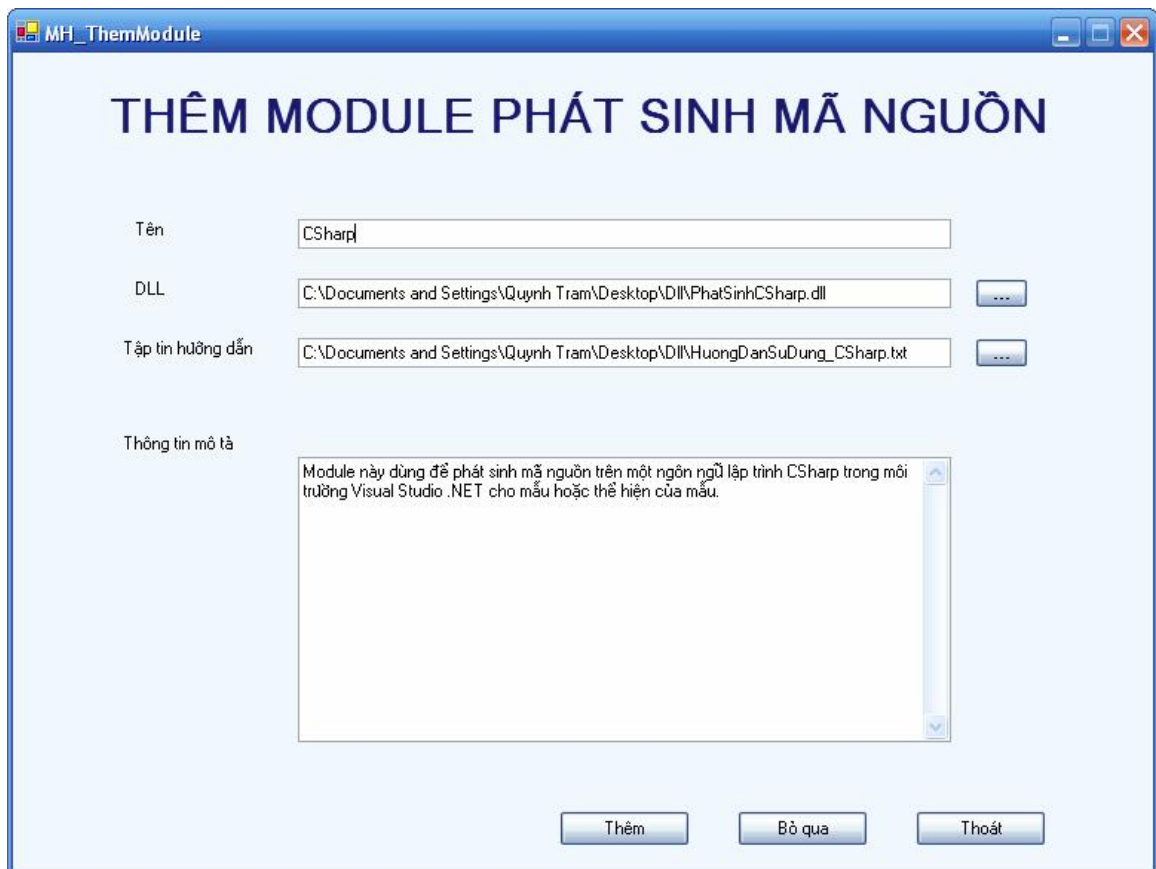
Mô tả cách sử dụng:

Bước 1: Chọn menu “Thể hiện mẫu” à “Nhập mã nguồn”.

Các bước sau tiến hành tương tự chức năng “Nhập mã nguồn” cho mẫu.

4.2.7 Quản lý module phát sinh mã nguồn

4.2.7.1 Thêm module phát sinh mã nguồn



THÊM MODULE PHÁT SINH MÃ NGUỒN

Tên: CSharp

DLL: C:\Documents and Settings\Quynh Tram\Desktop\Dll\PhatSinhCSharp.dll

Tập tin hướng dẫn: C:\Documents and Settings\Quynh Tram\Desktop\Dll\HuongDanSuDung_CSharp.txt

Thông tin mô tả: Module này dùng để phát sinh mã nguồn trên một ngôn ngữ lập trình CSharp trong môi trường Visual Studio .NET cho mẫu hoặc thể hiện của mẫu.

Thêm Bỏ qua Thoát

Hình 4-15 Màn hình thêm module phát sinh mã nguồn

Mô tả cách sử dụng:

Bước 1: Chọn menu “Ngôn ngữ lập trình” à “Phát sinh mã nguồn” à “Thêm”.

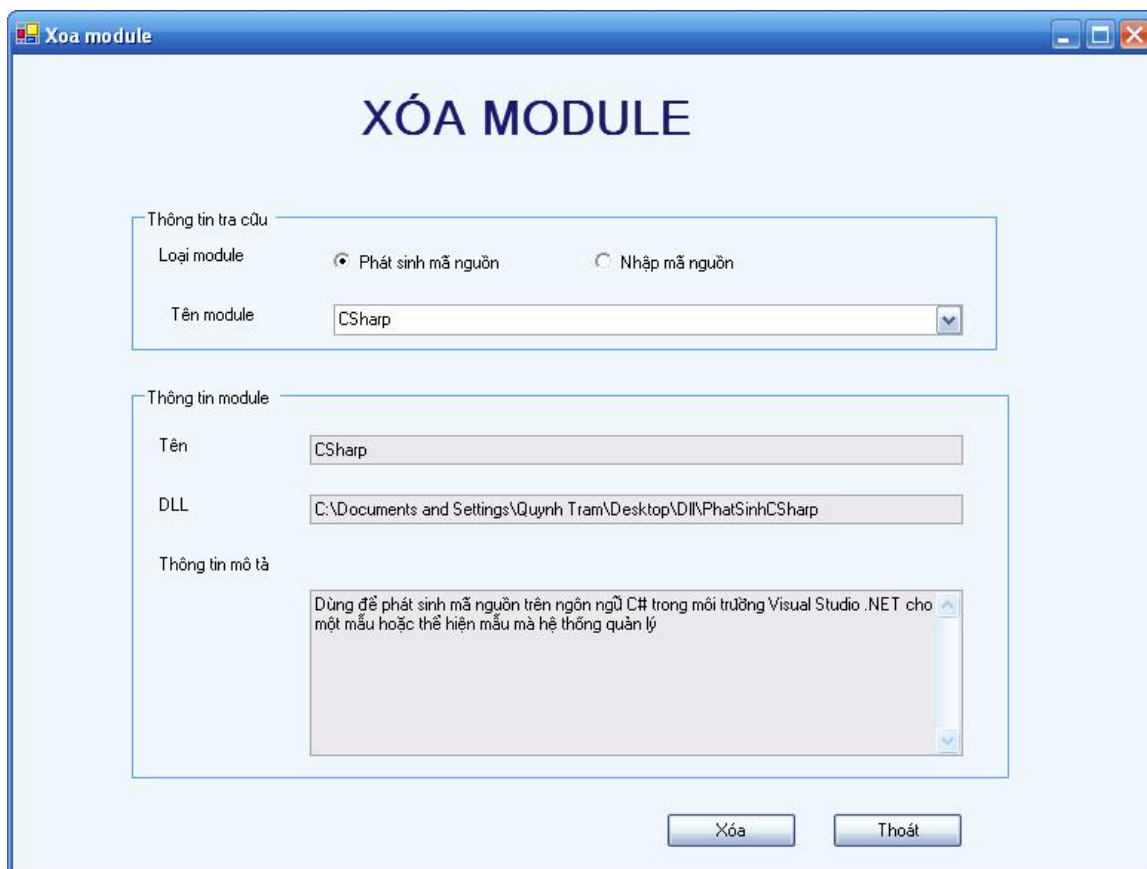
Bước 2: Nhập tên module, đường dẫn đến tập tin .dll của module, đường dẫn đến tập tin hướng dẫn sử dụng, thông tin mô tả.

Bước 3: Nhấn nút “Thêm”.

Ghi chú:

Nhấn nút “Bỏ qua” để xóa những thông tin vừa nhập trên màn hình.

4.2.7.2 Xóa module phát sinh mã nguồn



XÓA MODULE

Thông tin tra cứu

Loại module ☒ Phát sinh mã nguồn ☐ Nhập mã nguồn

Tên module CSharp

Thông tin module

Tên CSharp

DLL C:\Documents and Settings\Quynh Tram\Desktop\Dll\PhatSinhCSharp

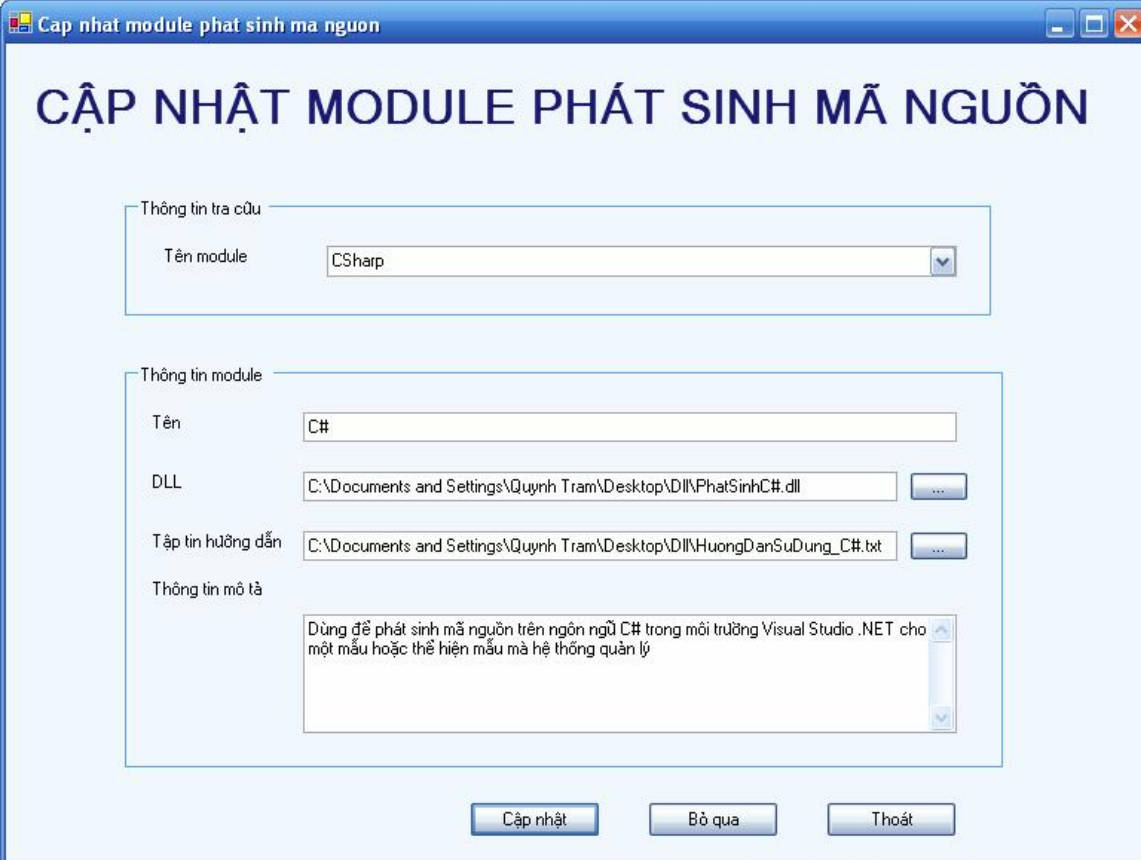
Thông tin mô tả

Dùng để phát sinh mã nguồn trên ngôn ngữ C# trong môi trường Visual Studio .NET cho một mẫu hoặc thể hiện mẫu mà hệ thống quản lý

Xóa Thoát

Hình 4-16 Màn hình xóa module phát sinh mã nguồn

4.2.7.3 Cập nhật module phát sinh mã nguồn



CẬP NHẬT MODULE PHÁT SINH MÃ NGUỒN

Thông tin tra cứu

Tên module: CSharp

Thông tin module

Tên: C#

DLL: C:\Documents and Settings\Quynh Tram\Desktop\Dll\PhatSinhC#.dll

Tập tin hướng dẫn: C:\Documents and Settings\Quynh Tram\Desktop\Dll\HuongDanSuDung_C#.txt

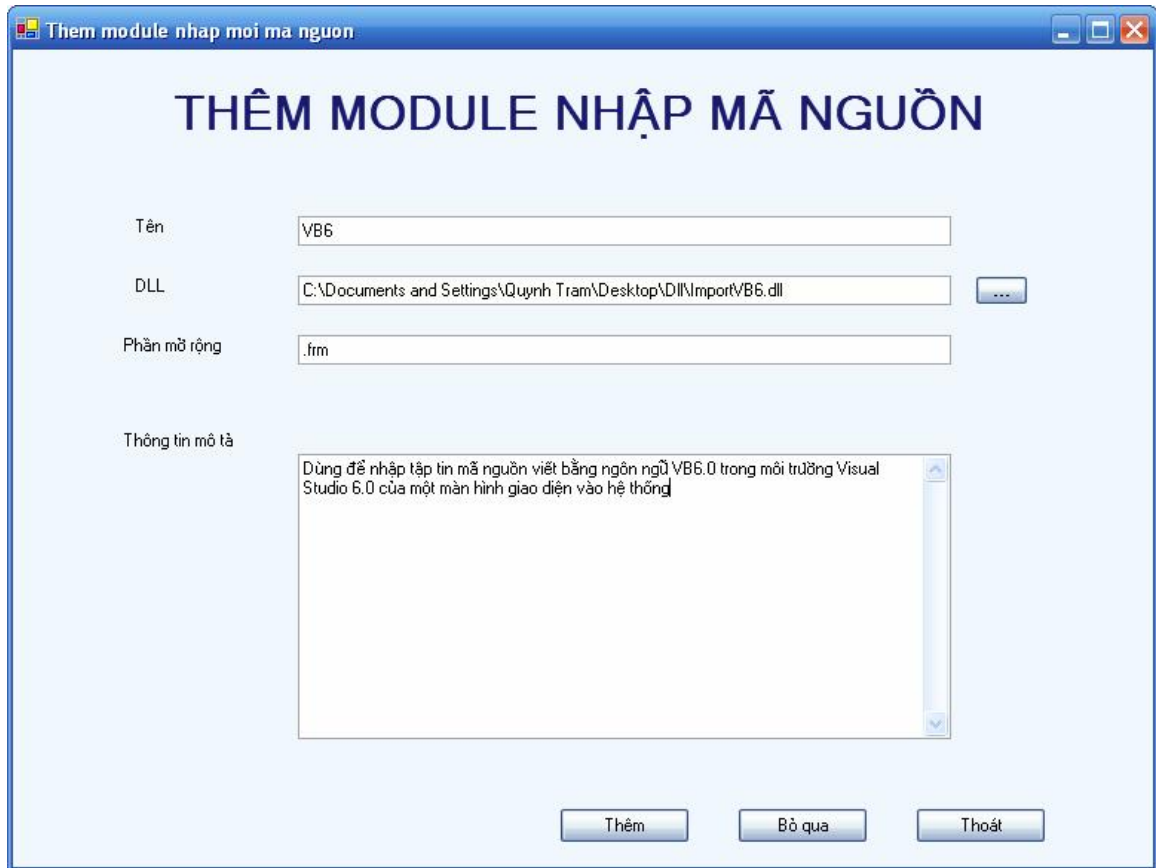
Thông tin mô tả: Dùng để phát sinh mã nguồn trên ngôn ngữ C# trong môi trường Visual Studio .NET cho một mẫu hoặc thể hiện mẫu mà hệ thống quản lý

Cập nhật Bỏ qua Thoát

Hình 4-17 Màn hình cập nhật module phát sinh mã nguồn

4.2.8 Quản lý module nhập từ mã nguồn

4.2.8.1 Thêm module nhập mã nguồn



Thêm module nhập mã nguồn

THÊM MODULE NHẬP MÃ NGUỒN

Tên: VB6

DLL: C:\Documents and Settings\Quynh Tram\Desktop\Dll\Import\VB6.dll

Phần mở rộng: .frm

Thông tin mô tả: Dùng để nhập tập tin mã nguồn viết bằng ngôn ngữ VB6.0 trong môi trường Visual Studio 6.0 của một màn hình giao diện vào hệ thống

Thêm Bỏ qua Thoát

Hình 4-18 Màn hình thêm module nhập mã nguồn

Mô tả cách sử dụng:

Bước 1: Chọn menu “Ngôn ngữ lập trình” à “Nhập mã nguồn”.

Bước 2: Nhập tên module, đường dẫn đến tập tin .dll của module, phần mở rộng của tập tin mã nguồn (ví dụ: đối với C# là .cs, VB6.0 là .frm), thông tin mô tả.

Bước 3: Nhấn nút “Thêm”.

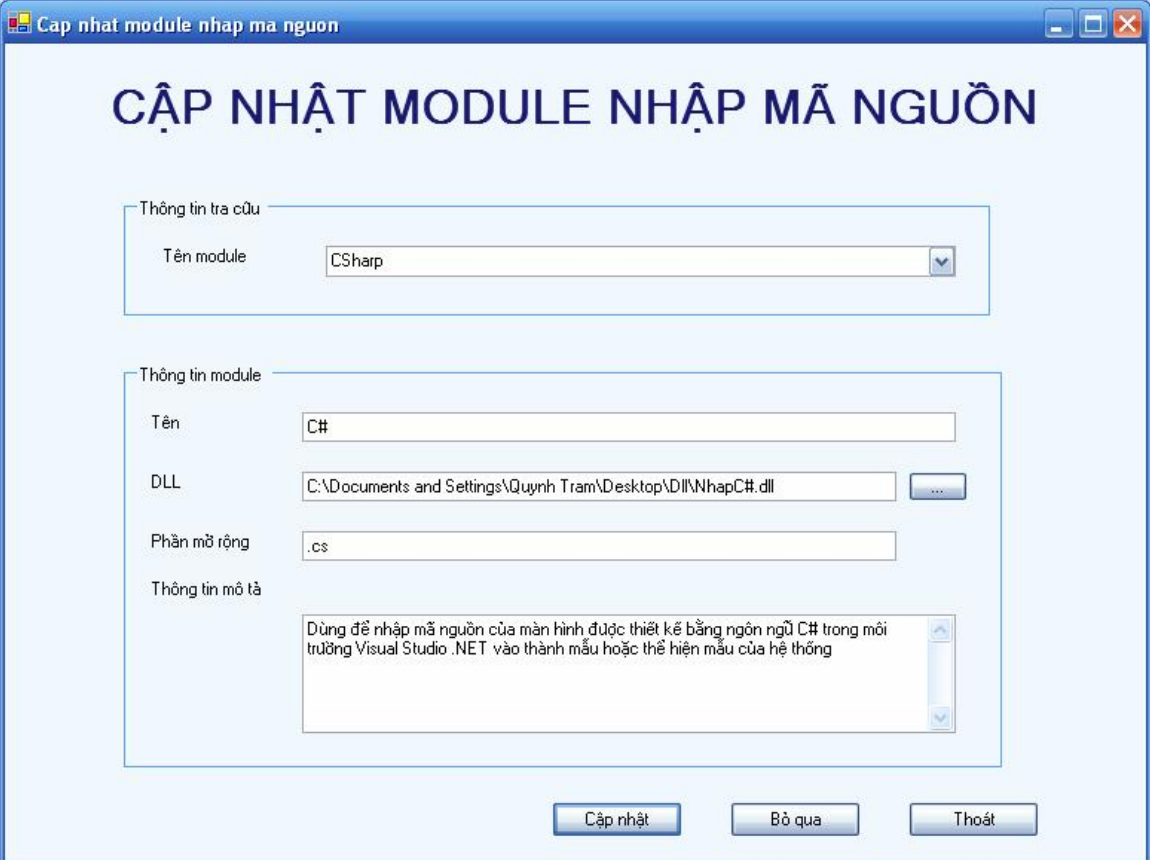
Ghi chú:

Có thể nhấn nút “Bỏ qua” để xóa các thông tin vừa nhập trên màn hình.

4.2.8.2 Xóa module nhập mã nguồn

Tương tự chức năng xóa module xóa module phát sinh mã nguồn.

4.2.8.3 Cập nhật module nhập mã nguồn



CẬP NHẬT MODULE NHẬP MÃ NGUỒN

Thông tin tra cứu

Tên module: CSharp

Thông tin module

Tên: C#

DLL: C:\Documents and Settings\Quynh Tram\Desktop\Dll\NhapC#.dll

Phần mở rộng: .cs

Thông tin mô tả: Dùng để nhập mã nguồn của màn hình được thiết kế bằng ngôn ngữ C# trong môi trường Visual Studio .NET vào thành mẫu hoặc thể hiện mẫu của hệ thống

Cập nhật Bỏ qua Thoát

Hình 4-19 Màn hình cập nhật module nhập mã nguồn

4.3 Thử nghiệm

4.3.1 Minh họa khả năng tái sử dụng các thành phần của hệ thống

Thử nghiệm:

Sau đây, chúng tôi tiến hành demo một số chức năng của hệ thống theo kịch bản sau: người sử dụng có nhu cầu lấy một thể hiện mẫu mà hệ thống lưu trữ, chỉnh sửa lại bằng cách thay đổi một số thành phần của thể hiện mẫu và đưa vào sử dụng trong ứng dụng mà họ đang phát triển.

Ý nghĩa của quá trình minh họa trên:

- Hệ thống đảm bảo được các chức năng đề ra.
- Khả năng tái sử dụng mà hệ thống hỗ trợ được thể hiện rõ nét: có thể **tái sử dụng một màn hình giao diện** mà hệ thống lưu trữ để sử dụng lại trong một ứng dụng mới. Hơn nữa, có thể **tái sử dụng ở mức độ từng thành phần** bộ phận của màn hình giao diện.
- Tập tin mã nguồn mà hệ thống hỗ trợ phát sinh có thể được sử dụng ngay trong ứng dụng mà họ đang phát triển.

Các bước mà người sử dụng sẽ tiến hành như sau:

Bước 1: Vào menu chính chọn “Thể hiện mẫu” à “Tra cứu”.

Sau bước này wizard hỗ trợ tra cứu thể hiện mẫu sẽ được hiển thị.

Bước 2: Tiến hành tra cứu theo các bước của wizard để được một thể hiện mẫu ưng ý nhất. Và nhấn nút “Hoàn tất”.

Sau bước này, thể hiện mẫu được chọn sẽ được hiển thị trong màn hình “Quản lý thể hiện mẫu”.

Nhận thấy, đây là thể hiện mẫu phục vụ cho việc xuất thông tin của nhân viên. Và nhu cầu lúc này là muốn có màn hình tra cứu nhân viên. Người sử dụng tiếp tục bước sau:

Bước 3: Gọi chức năng tra cứu thành phần.

Sau bước này wizard tra cứu thành phần được hiển thị.

Bước 4: Tiến hành chọn một bộ lọc ưng ý theo các bước mà wizard hỗ trợ. Sau đó nhấn nút “Hoàn tất”. Thành phần bộ lọc vừa chọn sẽ hiển thị trên màn hình “Quản lý thể hiện mẫu”.

Thay đổi vị trí bộ lọc cho phù hợp với thể hiện mẫu.

Bước 5: Gọi chức năng “Phát sinh mã nguồn” trên toolbar của màn hình “Quản lý thể hiện mẫu”. Màn hình hỗ trợ phát sinh mã nguồn được hiển thị.

Bước 6: Nhập những thông tin cần thiết để hệ thống phát sinh mã nguồn và nhấn nút “Phát sinh”. Cuối bước này, mã nguồn của thể hiện mẫu đã được lưu tại vị trí chỉ định, người sử dụng có thể thêm vào ứng dụng của mình và sử dụng ngay.

4.3.2 Các chức năng Phân quyền, quản lý hệ thống, thông tin cá nhân

Các chức năng được thử nghiệm :

- Đăng nhập lần lượt với 3 quyền Nhà quản trị, Nhà nghiên cứu mẫu và Người sử dụng mẫu để kiểm tra chức năng phân quyền.
- Thêm, Cập nhật, Xóa người dùng.
- Xem, Cập nhật thông tin cá nhân.

Kết quả: tốt.

4.3.3 Quản lý các module nhập vào mã nguồn và phát sinh mã nguồn

Các chức năng được thử nghiệm :

- Thêm, cập nhật thông tin của module, gỡ module ra khỏi hệ thống.
- Kiểm tra thông tin quản lý module ở các chức năng nhập vào mã nguồn và phát sinh giao diện cho mẫu/thể hiện mẫu.

Kết quả: tốt.

4.3.4 Công cụ đồ họa

Các chức năng được thử nghiệm :

- Thêm, xóa điều khiển trên màn hình đồ họa.
- Nhóm các điều khiển được chọn thành một thành phần, gỡ thành phần bao bên ngoài các điều khiển.

- Nhập vào màn hình đồ họa một thành phần đang được hệ thống quản lý.
- Lưu một thành phần trên màn hình đồ họa vào hệ thống để tái sử dụng về sau.

Kết quả: đáp ứng yêu cầu.

4.3.5 Quản lý thành phần

Các chức năng được thử nghiệm:

- Tạo mới một thành phần.
- Tạo một thành phần từ một thành phần đã có.
- Tra cứu thành phần.
- Chỉnh sửa giao diện của thành phần bằng công cụ đồ họa.
- Chỉnh sửa thông tin mô tả của thành phần.
- Thêm, cập nhật thành phần vào hệ thống, xóa thành phần khỏi hệ thống.
- Tạo biến thể cho thành phần.
- Mở đặc tả của mẫu từ tập tin xml và chuyển tiếp sang màn hình đồ họa.
- Lưu đặc tả của mẫu thành tập tin xml.

Kết quả: tốt.

4.3.6 Quản lý mẫu

Các chức năng được thử nghiệm:

- Tạo mới một mẫu.
- Tạo một mẫu từ một mẫu đã có.
- Nhập vào mã nguồn của mẫu và chuyển tiếp sang màn hình đồ họa.
- Tra cứu mẫu.
- Chỉnh sửa giao diện của mẫu bằng công cụ đồ họa.
- Chỉnh sửa thông tin mô tả của mẫu.
- Thêm, cập nhật mẫu vào hệ thống, xóa mẫu khỏi hệ thống.
- Tạo biến thể cho mẫu.
- Tạo thể hiện mẫu cho mẫu.
- Nhập một thành phần của hệ thống vào màn hình đồ họa của mẫu.

- Lưu một thành phần trên màn hình đồ họa của mẫu vào hệ thống.
- Mở đặc tả của mẫu từ tập tin xml và chuyển tiếp sang màn hình đồ họa.
- Lưu đặc tả của mẫu thành tập tin xml.

Kết quả: tốt.

4.3.7 Quản lý thể hiện mẫu

Các chức năng được thử nghiệm:

- Tạo mới một thể hiện mẫu.
- Tạo một thể hiện mẫu từ một mẫu đã có.
- Nhập vào mã nguồn của thể hiện mẫu và chuyển tiếp sang màn hình đồ họa.
- Tra cứu thể hiện mẫu.
- Chỉnh sửa giao diện của thể hiện mẫu bằng công cụ đồ họa.
- Chỉnh sửa thông tin mô tả của thể hiện mẫu.
- Thêm, cập nhật thể hiện mẫu vào hệ thống, xóa thể hiện mẫu khỏi hệ thống.
- Nhập một thành phần của hệ thống vào màn hình đồ họa của thể hiện mẫu.
- Lưu một thành phần trên màn hình đồ họa của thể hiện mẫu vào hệ thống.
- Mở đặc tả của thể hiện mẫu từ tập tin xml và chuyển tiếp sang màn hình đồ họa.
- Lưu đặc tả của thể hiện mẫu thành tập tin xml.

Kết quả: tốt.

Chương 5 Tổng kết và hướng phát triển

Nội dung

Chương này sẽ trình bày kết quả đạt được của luận văn cùng những hạn chế, từ đó đề ra hướng phát triển trong tương lai.

KHOA CNTT

5.1 Kết quả đạt được

- Về Lý thuyết:

- Đã tìm hiểu được hai loại mẫu thiết kế giao diện phổ biến nhất trong hệ thống thông tin, đó là loại mẫu Simple và loại mẫu Master Detail. Ở mỗi loại mẫu, chúng tôi đã nắm vững ý nghĩa và cách ứng dụng chúng trong hệ thống thông tin.
- Hiểu được ý nghĩa và vai trò của hệ thống quản lý mẫu thiết kế và thành phần từ mẫu thiết kế phục vụ cho việc xây dựng hệ thống thông tin.
- Nắm được nội dung và một số trường hợp ứng dụng các mẫu thiết kế Composite và Proxy vào một ứng dụng cụ thể.

- Về Kỹ thuật:

- Kỹ năng lập trình theo mô hình 3 tầng.
- Kỹ năng lập trình trên môi trường .Net, cụ thể là ngôn ngữ lập trình C# với bộ thư viện đồ họa GDI+.
- Vận dụng thành công mẫu thiết kế Composite và Proxy để hỗ trợ công cụ đồ họa và khả năng plug-in các module vào hệ thống chính.
- Vận dụng kết hợp cơ sở dữ liệu quan hệ và Xml trong để giải quyết việc lưu trữ một loại dữ liệu đặc biệt là mẫu thiết kế giao diện.
- Kỹ năng xử lý đặc tả và mã nguồn để hỗ trợ chức năng nhập vào mã nguồn của mẫu/thể hiện mẫu cũng như phát sinh giao diện cho mẫu/thể hiện mẫu.

- Về Thực nghiệm:

- Xây dựng thành công hệ thống quản lý mẫu thiết kế giao diện và các thành phần từ mẫu thiết kế giao diện với các chức năng đã đề ra. Hiện tại hệ thống quản lý hai loại mẫu Simple và Master Detail,

cho phép nhập vào mã nguồn và phát sinh giao diện trên 2 ngôn ngữ lập trình VB 6.0 và C#.

- Khả năng phân quyền: đạt yêu cầu phân quyền cho nhà nghiên cứu mẫu và người sử dụng mẫu.
- Hệ thống có tính tiến hóa cao: linh hoạt trong việc mở rộng số lượng điều khiển trong giao diện, cài đặt hay gỡ bỏ các module nhập vào mã nguồn của mẫu/thể hiện mẫu và phát sinh giao diện cho mẫu/thể hiện mẫu.

5.2 Hạn chế

Qua thử nghiệm, chúng tôi nhận thấy có những trường hợp người sử dụng mong muốn mọi thay đổi trên thành phần ban đầu sẽ không làm ảnh hưởng đến các mẫu/thể hiện mẫu có sử dụng chúng. Các mẫu/thể hiện mẫu này thường đã đạt yêu cầu và được sử dụng thành công. Bên cạnh đó, đối với những mẫu/thể hiện mẫu còn đang thử nghiệm, người sử dụng lại mong muốn hệ thống tự động cập nhật chúng theo các thành phần mà chúng có sử dụng, thay vì phải thay đổi thủ công từng trường hợp một.

Hiện tại, sau khi lắp ghép các thành phần để tạo ra mẫu/thể hiện mẫu và lưu trữ mẫu/thể hiện mẫu này, mọi cập nhật nếu có trên thành phần không được cập nhật trên mẫu/thể hiện mẫu.

5.3 Hướng phát triển

- Khắc phục hạn chế bằng cách cho phép người dùng quyết định xem mẫu/thể hiện mẫu của mình có được cập nhật theo thành phần mà nó sử dụng hay không.
- Tìm hiểu và đưa vào hệ thống các mẫu thiết kế giao diện khác.
- Hỗ trợ thêm các điều khiển khác ngoài những điều khiển chuẩn đã có để tăng tính phong phú cho hệ thống.

- Mở rộng khả năng hỗ trợ nhập vào mã nguồn của mẫu/thể hiện mẫu cũng như phát sinh giao diện cho mẫu/thể hiện mẫu trên các ngôn ngữ lập trình khác.
- Hệ thống chỉ hỗ trợ người dùng thiết kế thuộc tính đơn giản cho các điều khiển. Tuy nhiên, một số điều khiển như listview, grid hệ thống chưa hỗ trợ thiết kế thuộc tính của các điều khiển này theo thuộc tính của các đối tượng dữ liệu được lưu trữ. Ví dụ: dùng grid để hiển thị thông tin sinh viên, nhưng người dùng chưa thể thiết kế các cột của grid theo các trường của bảng sinh viên. Vì vậy, hướng phát triển trong tương lai là sẽ hỗ trợ chức năng trên.

KHOA CNTT

Phụ lục

Chi tiết các lớp xử lý chính

ThanhPhan	
xl_ThanhPhan	: XL_ThanhPhan
ma	: int
maCha	: int
ten	: int
tacGia	: string
ngayTao	: string
ngayCNCuoi	: string
xDacTa	: XmlNode
hinhAnh	: Image
myComponent	: MyComponent
DocXDacTa()	: void
TaoXDacTa (XmlDocument)	: void
TimTheoTen(string tenCanTim)	: bool
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
Ve()	: void
LayViTriDiemBao()	: int
LayTatCaDieuKhien()	: ArrayList
LayDieuKhien(int i)	: MyControl
BoDieuKhienChon()	: void
ThemDieuKhien(MyControl dk)	: bool
XoaDieuKhienChon()	: void
DiChuyenDieuKhienChon()	: void
TDKichThuocDieuKhienChon()	: void
KTViTriHopLe()	: bool
KTTrungTen()	: bool

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	xl_ThanhPhan	XL_ThanhPhan	Đối tượng lớp LT_ThanhPhan	
2	Ma	Int	Đối tượng lớp ThanhPhan	
3	maCha	Int	Mã cha	
4	Ten	String	Tên	
5	tacGia	String	Tác giả	
6	ngayTao	String	Ngày tạo	
7	ngayCNCuoi	String	Ngày cập nhật cuối	
8	xDacTa	XmlNode	Đặc tả Xml	
9	hinhAnh	Image	Hình ảnh minh họa	
10	MyComponent	MyComponent	Đối tượng lớp MyComponent	

Bảng PL - 1 Danh sách thuộc tính lớp ThanhPhan

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	DocXDacTa		void	Đọc đặc tả xml từ thuộc tính xDacTa sang các thuộc tính khác	
2	TaoXDacTa	XmlDocument xDoc	void	Chuyển các thuộc tính thành chuỗi Xml và đưa vào thuộc tính xDacTa	
3	TimTheoTen	String tenCanTim	bool	Tìm thành phần theo tên	
4	Them		bool	Thêm thành phần vào cơ sở dữ liệu	
5	CapNhat		bool	Thêm thành phần từ cơ sở dữ liệu	
6	Xoa		void	Xóa thành phần từ cơ sở dữ liệu	
7	Ve		void	Vẽ	

8	LayViTriDiemBao		int	Lấy vị trí điểm bao	
9	LayTatCaDieuKien		ArrayList	Lấy tất cả điều khiển	
10	LayDieuKien	int i	MyControl	Lấy điều khiển	
11	BoDieuKien Chon		void	Bỏ điều khiển chọn	
12	ThemDieuKien	MyControl dk	bool	Thêm điều khiển	
13	XoaDieuKienChon		void	Xóa điều khiển được chọn	
14	DiChuyenDieuKien Chon		void	Di chuyển điều khiển được chọn	
15	TDKichThuoc DieuKienChon		void	Thay đổi kích thước các điều khiển được chọn	
16	KTViTriHopLe		bool	Kiểm tra vị trí hợp lệ	
17	KTTrungTen		bool	Kiểm tra trùng tên	

Bảng PL - 2 Danh sách hàm lớp **ThanhPhan**

XL_ThanhPhan	
lt_ThanhPhan	: LT_ThanhPhan
thanhPhan	: ThanhPhan
TimTheoTen(string tenCanTim) : bool Them() : bool CapNhat() : bool Xoa() : bool LayDS() : ThanhPhan[] LayTheoMa(int ma) : ThanhPhan TaoXDacTa(XmlDocument xDoc) : void ThôngTinMota2Xml() : void GiaoDien2Xml() :void DocXDacTa() :void Xml2ThôngTinMoTa() :void Xml2GiaoDien() :void	

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	lt_ThanhPhan	LT_ThanhPhan	Đối tượng lớp LT_ThanhPhan	
2	thanhPhan	ThanhPHan	Đối tượng lớp ThanhPhan	

Bảng PL - 3 Danh sách thuộc tính lớp XL_ThanhPhan

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	TimTheoTen		bool	Tìm the tên	
2	Them		bool	Thêm vào cơ sở dữ liệu	
3	CapNhat		bool	Cập nhật	
4	Xoa		bool	Xóa	
5	LayDS		ThanhPhan[]	Lấy danh sách các thành phần	
6	LayTheoMa	int ma	ThanhPhan	Lấy thành phần theo mã	
7	TaoXDacTa	XmlDocument xDoc	void	Tạo xDacTa từ các thuộc tính	
8	ThongTin Mota2Xml		void	Chuyển thông tin mô tả thành chuỗi XML	
9	GiaoDien2Xml		void	Chuyển các điều khiển con thành Xml	
10	DocXDacTa		void	Đọc xDacTa	
11	Xml2 ThongTin MoTa		void	Chuyển chuỗi Xml thành thông tin mô tả	
12	Xml2 GiaoDien		bool	Chuyển chuỗi Xml thành danh sách các điều khiển	

Bảng PL - 4 Danh sách hàm lớp XL_ThanhPhan

LT_ThanhPhan	
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
LayDS()	: ThanhPhan[]
LayTheoMa(int ma)	: ThanhPhan

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	Them		bool	Thêm vào cơ sở dữ liệu	
2	CapNhat		bool	Cập nhật	
3	Xoa		bool	Xóa	
4	LayDS		ThanhPhan[]	Lấy danh sách các thành phần	
5	LayTheoMa	int ma	ThanhPhan	Lấy thành phần theo mã	

Bảng PL - 5 Danh sách hàm lớp LT_ThanhPhan

Mau	
xl_Mau	: XL_Mau
ma	: int
maCha	: int
ten	: int
tacGia	: string
ngayTao	: string
ngayCNCuoi	: string
xDacTa	: XmlNode
hinhAnh	: Image
manHinh	: MyForm
DocXDacTa()	: void
TaoXDacTa (XmlDocument)	: void

TimTheoTen(string tenCanTim) : bool
 Them() : bool
 CapNhat() : bool
 Xoa() : bool
 Ve() : void
 LayViTriDiemBao() : int
 LayTatCaDieuKhien() : ArrayList
 LayDieuKhien(int i) : MyControl
 BoDieuKhienChon() : void
 ThemDieuKhien(MyControl dk) : bool
 XoaDieuKhienChon() : void
 DiChuyenDieuKhienChon() : void
 TDKichThuocDieuKhienChon() : void
 KTViTriHopLe() : bool
 KTTrungTen() : bool

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	xl_Mau	XL_Mau	Đối tượng lớp LT_Mau	
2	Ma	Int	Đối tượng lớp Mau	
3	maCha	Int	Mã cha	
4	Ten	String	Tên	
5	tacGia	String	Tác giả	
6	ngayTao	String	Ngày tạo	
7	ngayCNCuoi	String	Ngày cập nhật cuối	
8	xDacTa	XmlNode	Đặc tả Xml	
9	hinhAnh	Image	Hình ảnh minh họa	
10	manHinh	MyForm	Đối tượng lớp MyForm	

Bảng PL - 6 Danh sách thuộc tính lớp Mau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	DocXDacTa		void	Đọc đặc tả xml từ thuộc tính xDacTa sang các thuộc tính khác	
2	TaoXDacTa	XmlDocument xDoc	void	Chuyển các thuộc thành chuỗi Xml và đưa vào thuộc tính xDacTa	
3	TimTheoTen	String tenCanTim	bool	Tìm mẫu theo tên	
4	Them		bool	Thêm mẫu vào cơ sở dữ liệu	
5	CapNhat		bool	Thêm mẫu từ cơ sở dữ liệu	
6	Xoa		void	Xóa mẫu từ cơ sở dữ liệu	
7	Ve		void	Vẽ	
8	LayViTriDiemBao		int	Lấy vị trí điểm bao	
9	LayTatCaDieuKhen		ArrayList	Lấy tất cả điều khiển	
10	LayDieuKhen	int i	MyControl	Lấy điều khiển	
11	BoDieuKhenChon		void	Bỏ điều khiển chọn	
12	ThemDieuKhen	MyControl dk	bool	Thêm điều khiển	
13	XoaDieuKhenChon		void	Xóa điều khiển được chọn	
14	DiChuyenDieuKhenChon		void	Di chuyển điều khiển được chọn	
15	TDKichThuocDieuKhenChon		void	Thay đổi kích thước các điều khiển được chọn	
16	KTViTriHopLe		bool	Kiểm tra vị trí hợp lệ	
17	KTTrungTen		bool	Kiểm tra trùng tên	

Bảng PL - 7 Danh sách hàm lớp Mau

XL_Mau	
lt_Mau	: LT_Mau
mau	: Mau
TimTheoTen(string tenCanTim) : bool Them() : bool CapNhat() : bool Xoa() : bool LayDS() : Mau[] LayTheoMa(int ma) : Mau TaoXDacTa(XmlDocument xDoc) : void ThôngTinMota2Xml() : void GiaoDien2Xml() :void DocXDacTa() :void Xml2ThôngTinMoTa() :void Xml2GiaoDien() :void	

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	lt_Mau	LT_Mau	Đối tượng lớp LT_Mau	
2	Mau	Mau	Đối tượng lớp Mau	

Bảng PL - 8 Danh sách thuộc tính lớp XL_Mau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	TimTheoTen		bool	Tìm the tên	
2	Them		bool	Thêm vào cơ sở dữ liệu	
3	CapNhat		bool	Cập nhật	
4	Xoa		bool	Xóa	
5	LayDS		Mau[]	Lấy danh sách các mẫu	
6	LayTheoMa	int ma	Mau	Lấy mẫu theo mã	
7	TaoXDacTa	XmlDocument xDoc	void	Tạo xDacTa từ các thuộc tính	

8	ThôngTin Mota2Xml		void	Chuyển thông tin mô tả thành chuỗi XML	
9	GiaoDien2Xml		void	Chuyển các điều khiển con thành Xml	
10	DocXDacTa		void	Đọc xDacTa	
11	Xml2 ThôngTin MoTa		void	Chuyển chuỗi Xml thành thông tin mô tả	
12	Xml2 GiaoDien		bool	Chuyển chuỗi Xml thành danh sách các điều khiển	

Bảng PL - 9 Danh sách hàm lớp XL_Mau

LT_Mau	
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
LayDS()	: Mau[]
LayTheoMa(int ma)	: Mau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	Them		bool	Thêm vào cơ sở dữ liệu	
2	CapNhat		bool	Cập nhật	
3	Xoa		bool	Xóa	
4	LayDS		Mau[]	Lấy danh sách các mẫu	
5	LayTheoMa	int ma	Mau	Lấy mẫu theo mã	

Bảng PL - 10 Danh sách hàm lớp LT_Mau

TheHienMau	
xl_TheHienMau	: XL_TheHienMau
ma	: int
maMau	: int
ten	: int
tacGia	: string
ngayTao	: string
ngayCNCuoi	: string
nguCanhSuDung	: string
xDacTa	: XmlNode
hinhAnh	: Image
manHinh	: MyForm
DocXDacTa()	: void
TaoXDacTa (XmlDocument)	: void
TimTheoTen(string tenCanTim)	: bool
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
Ve()	: void
LayViTriDiemBao()	: int
LayTatCaDieuKhien()	: ArrayList
LayDieuKhien(int i)	: MyControl
BoDieuKhienChon()	: void
ThemDieuKhien(MyControl dk)	: bool
XoaDieuKhienChon()	: void
DiChuyenDieuKhienChon()	: void
TDKichThuocDieuKhienChon()	: void
KTViTriHopLe()	: bool
KTTrungTen()	: bool

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	xl_TheHienMau	XL_TheHienMau	Đối tượng lớp LT_TheHienMau	
2	Ma	Int	Đối tượng lớp TheHienMau	
3	maMau	Int	Mã mẫu gốc	
4	Ten	String	Tên	
5	tacGia	String	Tác giả	
6	ngayTao	String	Ngày tạo	
7	ngayCNCuoi	String	Ngày cập nhật cuối	
8	xDacTa	XmlNode	Đặc tả Xml	
8	nguCanhSuDung	String	Ngữ cảnh sử dụng	
10	hinhAnh	Image	Hình ảnh minh họa	
11	manHinh	MyForm	Đối tượng lớp MyForm	

Bảng PL - 11 Danh sách thuộc tính lớp TheHienMau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	DocXDacTa		void	Đọc đặc tả xml từ thuộc tính xDacTa sang các thuộc tính khác	
2	TaoXDacTa	XmlDocument xDoc	void	Chuyển các thuộc tính thành chuỗi Xml và đưa vào thuộc tính xDacTa	
3	TimTheoTen	String tenCanTim	bool	Tìm thể hiện mẫu theo tên	
4	Them		bool	Thêm thể hiện mẫu vào cơ sở dữ liệu	
5	CapNhat		bool	Thêm thể hiện mẫu từ cơ sở dữ liệu	
6	Xoa		void	Xóa thể hiện mẫu từ cơ sở dữ liệu	
7	Ve		void	Vẽ	

8	LayViTriDiemBao		int	Lấy vị trí điểm bao	
9	LayTatCaDieuKhien		ArrayList	Lấy tất cả điều khiển	
10	LayDieuKhien	int i	MyControl	Lấy điều khiển	
11	BoDieuKhien Chon		void	Bỏ điều khiển chọn	
12	ThemDieu Khien	MyControl dk	bool	Thêm điều khiển	
13	XoaDieuKhienChon		void	Xóa điều khiển được chọn	
14	DiChuyenDieuKhien Chon		void	Di chuyển điều khiển được chọn	
15	TDKichThuoc DieuKhienChon		void	Thay đổi kích thước các điều khiển được chọn	
16	KTViTriHopLe		bool	Kiểm tra vị trí hợp lệ	
17	KTTrungTen		bool	Kiểm tra trùng tên	

Bảng PL - 12 Danh sách hàm lớp **TheHienMau**

XL_TheHienMau	
lt_TheHienMau	: LT_TheHienMau
mau	: TheHienMau
TimTheoTen(string tenCanTim)	: bool
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
LayDS()	: TheHienMau[]
LayTheoMa(int ma)	: TheHienMau
TaoXDacTa(XmlDocument xDoc)	: void
ThongTinMota2Xml()	: void
GiaoDien2Xml()	:void
DocXDacTa()	:void
Xml2ThongTinMoTa()	:void
Xml2GiaoDien()	:void

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	lt_TheHienMau	LT_TheHienMau	Đối tượng lớp LT_TheHienMau	
2	TheHienMau	TheHienMau	Đối tượng lớp TheHienMau	

Bảng PL - 13 Danh sách thuộc tính lớp XL_TheHienMau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	TimTheoTen		bool	Tìm the tên	
2	Them		bool	Thêm vào cơ sở dữ liệu	
3	CapNhat		bool	Cập nhật	
4	Xoa		bool	Xóa	
5	LayDS		TheHienMau[]	Lấy danh sách các thể hiện mẫu	
6	LayTheoMa	int ma	TheHienMau	Lấy thể hiện mẫu theo mã	
7	TaoXDacTa	XmlDocument xDoc	void	Tạo xDacTa từ các thuộc tính	
8	ThongTinMota2Xml		void	Chuyển thông tin mô tả thành chuỗi XML	
9	GiaoDien2Xml		void	Chuyển các điều khiển con thành Xml	
10	DocXDacTa		void	Đọc xDacTa	
11	Xml2ThongTinMoTa		void	Chuyển chuỗi Xml thành thông tin mô tả	
12	Xml2GiaoDien		bool	Chuyển chuỗi Xml thành danh sách các điều khiển	

Bảng PL - 14 Danh sách hàm XL_TheHienMau

LT_TheHienMau	
Them()	: bool
CapNhat()	: bool
Xoa()	: bool
LayDS()	: TheHienMau[]
LayTheoMa(int ma)	: TheHienMau

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	Them		bool	Thêm vào cơ sở dữ liệu	
2	CapNhat		bool	Cập nhật	
3	Xoa		bool	Xóa	
4	LayDS		TheHienMau []	Lấy danh sách các thẻ hiện mẫu	
5	LayTheoMa	int ma	TheHienMau	Lấy thẻ hiện mẫu theo mã	

Bảng PL - 15 Danh sách thuộc tính lớp LT_TheHienMau

MyControl	
tenDieuKhien	: string
xDacTa	: XmlElement
khungDiChuyen	: Rectangle
dsDiemBao	: Point[]
duocChon	: bool
control	: Control
dkCha	: MyControl
coDinhChieuDai	: bool
coDinhChieuRong	: bool
SaoChep() : MyControl	
TaoDacTaTuXml(XmlDocument xDoc) : void	
KhoiDongThuocTinhTuXml() : void	
KhoiDongThuocTinhTuMaNguon_CSharp(ArrayList manguon) : void	

KhoiDongThuocTinhTuMaNguon_VB(ArrayList manguon) : void
 Ve(Graphics gr) : void
 ThayDoiKichThuoc(Point delta, int xContro, int yContro) : void
 KTKichThuocHopLe(Rectangle hcn) : bool
 ThemDieuKien(MyControl dk)
 LaySoDieuKien() : int
 KTTrungTen(MyControl dkKiemTra) : bool

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	tenDieuKien	string	Tên điều khiển	
2	xDacTa	XmlElement	Phần tử xml đặc tả thông tin riêng và tất cả các điều khiển thành phần	
3	khungDiChuyen	Rectangle	Khung vẽ khi di chuyển điều khiển trên màn hình đồ họa	
4	dsDiemBao	Point[]	8 điểm bao phục vụ việc thay đổi kích thước điều khiển	
5	duocChon	bool	Cờ được chọn, cho biết người dùng có chọn điều khiển này hay không	
6	control	Control	Đối tượng điều khiển	
7	dkCha	MyControl	Điều khiển cha	
8	coDinhChieuDai	bool	Cờ cố định chiều dài	
9	coDinhChieuRong	bool	Cờ cố định chiều rộng	

Bảng PL - 16 Danh sách các thuộc tính lớp MyControl

ST T	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	TaoDacTaTuXml	XmlDocument xDoc	void	Chuyển các thuộc tính thành chuỗi Xml	Hàm ảo
2	KhoiDongThuocTinhTu Xml		void	Khởi động các thuộc tính từ chuỗi Xml	Hàm ảo
3	KhoiDongThuocTinhTu MaNguon_CSharp	ArrayList manguon	void	Khởi động thuộc tính từ tập các dòng lệnh Csharp	Hàm ảo
4	KhoiDongThuocTinhTu MaNguon_VB	ArrayList manguon	void	Khởi động thuộc tính từ các dòng lệnh VB 6.0	Hàm ảo
5	Ve	Graphics gr	void	Vẽ điều khiển lên màn hình đồ họa	Hàm ảo
6	ThayDoiKichThuoc	Point delta, int xContro, int yContro	void	Thay đổi kích thước điều khiển	
7	KTKichThuocHopLe	Rectangle hcn	bool	Kiểm tra kích thước hợp lệ	
8	ThemDieuKhien	MyControl dk	void	Thêm điều khiển con	Hàm ảo
9	LaySoDieuKhien		int	Lấy số điều khiển con	Hàm ảo
10	KTTrungTen	MyControl dkKiemTra	bool	Kiểm tra có trùng tên với 1 điều khiển khác	

Bảng PL - 17 Danh sách hàm lớp MyControl

Các lớp MyLabel, MyTextBox, MyButton, MyRadioButton, MyCheckBox, MyGrid, MyListBox, MyComboBox, MyListView và MyTreeView đều có chi tiết tương tự, ngoại trừ các thuộc tính riêng. Chúng tôi chỉ trình bày chi tiết đại diện một lớp, đó là MyLabel.

MyLabel : MyControl	
Chỉ sử dụng các thuộc tính của control đã khai báo ở lớp MyControl	
MyLabel(int tab) TaoDacTaXml() : void KhoiDongThuocTinhTuXml(XmlDocument xDoc) : voi KhoiDongThuocTinhTuMaNguon_CSharp(ArrayList manguon) : void KhoiDongThuocTinhTuMaNguon_VB(ArrayList manguon) :void Ve(Graphics gr) : void	

ST T	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	MyLabel	int tab		Thiết lập đối tượng với giá trị tab	
2	TaoDacTaTuXml	XmlDocument xDoc	void	Chuyển các thuộc tính thành chuỗi Xml	Hàm nạp chồng
3	KhoiDongThuocTinhTuXml		void	Khởi động các thuộc tính từ chuỗi Xml	Hàm nạp chồng
4	KhoiDongThuocTinhTuMaNguon_CSharp	ArrayList manguon	void	Khởi động thuộc tính từ tập các dòng lệnh Csharp	Hàm nạp chồng
5	KhoiDongThuocTinhTuMaNguon_VB	ArrayList manguon	void	Khởi động thuộc tính từ các dòng lệnh VB 6.0	Hàm nạp chồng
6	Ve	Graphics gr	void	Vẽ điều khiển lên màn hình đồ họa	Hàm nạp chồng

Bảng PL - 18 Danh sách hàm lớp MyLabel

Các lớp kế thừa từ lớp MyComposite bao gồm MyComponent, MyForm và MyGroupBox. Trong đó MyComponent đại diện cho Thành phần trên màn hình đồ họa, MyForm đại diện cho màn hình của Mẫu và Thẻ hiện mẫu trên màn hình đồ họa, còn MyGroupBox chính là điều khiển GroupBox.

Cấu trúc của các lớp vừa liệt kê đều tương tự nhau, chỉ khác về thuộc tính và phạm vi ứng dụng. Chúng tôi xin trình bày đại diện một lớp, đó là lớp MyComposite.

MyComposite : MyControl	
dsDieuKien : ArrayList	
MyComposite(int tab)	
TaoDacTaXml() : void	
KhoiDongThuocTinhTuXml(XmlDocument xDoc) : voi	
KhoiDongThuocTinhTuMaNgon_CSharp(ArrayList manguon) : void	
KhoiDongThuocTinhTuMaNgon_VB(ArrayList manguon) :void	
Ve(Graphics gr) : void	

STT	Tên	Kiểu / Lớp	Ý nghĩa	Ghi chú
1	dsDieuKien	ArrayList	Danh sách các điều khiển con	

Bảng PL - 19 Danh sách thuộc tính lớp MyComposite

STT	Tên	Tham số	Kết quả	Xử lý	Ghi chú
1	MyComposite	int tab		Thiết lập đối tượng với giá trị tab	
2	TaoDacTaTuXml	Xml Document xDoc	void	Chuyển các thuộc tính của nó và cả danh sách điều khiển con thành chuỗi Xml	Hàm nạp chồng
3	KhoiDongThuocTinhTuXml		void	Khởi động các thuộc tính của nó và của các điều khiển con từ chuỗi Xml	Hàm nạp chồng
4	KhoiDongThuocTinhTuMaNgon_CSharp	ArrayList manguon	void	Khởi động thuộc tính và các điều khiển con từ tập các dòng lệnh CSharp	Hàm nạp chôn
5	KhoiDongThuocTinhTuMaNgon_VB	ArrayList manguon	void	Khởi động thuộc tính và các điều khiển con từ các dòng lệnh VB 6.0	Hàm nạp chồng
6	Ve	Graphics gr	void	Vẽ chính nó và các điều khiển con lên màn hình đồ họa	Hàm nạp chồng

Bảng PL - 20 Danh sách hàm lớp MyComposite

Chi tiết các bảng trong cơ sở dữ liệu:

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int		Mã	
2	Ten	nvarchar(50)		Tên	

Bảng PL - 21 Chi tiết bảng Quyen

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int	Khóa chính	Mã	
2	TenDN	nvarchar(50)		Tên đăng nhập	
3	MatKhau	nvarchar(50)		Mật khẩu	
4	Ten	nvarchar(50)		Tên	
5	MaQuyen	Int	Khóa ngoại	Tham chiếu đến trường Ma của bảng Quyen	

Bảng PL - 22 Chi tiết bảng NguoIDung

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int	Khóa chính	Mã	
2	Ten	nvarchar(50)		Tên	
3	ThongTin	nvarchar(50)		Thông tin	

Bảng PL - 23 Chi tiết bảng LoaiMau

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int		Mã	
2	Ten	nvarchar(50)		Tên	

Bảng PL - 24 Chi tiết bảng NguoIDung

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	int	Khóa chính	Mã	
2	MaCha	int	Khóa ngoại	Tham chiếu đến trường Mã của chính bảng này	
3	Ten	nvarchar(50)		Tên	
4	TacGia	nvarchar(50)		Tác giả	
5	NgayTao	datetime		Ngày tạo	
6	NgayCNCuoi	datetime		Ngày cập nhật cuối	
7	NoiDungXml	ntext		Nội dung dưới dạng xml	
8	HinhAnh	nvarchar(50)		Đường dẫn tập tin hình ảnh	

Bảng PL - 25 Chi tiết bảng ThanhPhan

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	int	Khóa chính	Mã	
2	MaCha	int	Khóa ngoại	Tham chiếu đến trường Mã của chính bảng này	
3	MaLoaiMau	int	Khóa ngoại	Tham chiếu đến trường Mã của bảng LoaiMau	
4	MaLoaiGiaoDien	int	Khóa ngoại	Tham chiếu đến trường Mã của bảng LoaiGiaoDien	
5	Ten	nvarchar(50)		Tên	
6	TacGia	nvarchar(50)		Tác giả	
7	NgayTao	datetime		Ngày tạo	
8	NgayCNCuoi	datetime		Ngày cập nhật cuối	
9	NoiDungXml	ntext		Nội dung dưới dạng xml	
10	HinhAnh	nvarchar(50)		Đường dẫn tập tin hình ảnh	

Bảng PL - 26 Chi tiết bảng Mẫu

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	int	Khóa chính	Mã	
2	MaMau	int	Khóa ngoại	Tham chiếu đến trường Mã của bảng Mau	
3	Ten	nvarchar(50)		Tên	
4	TacGia	nvarchar(50)		Tác giả	
5	NgayTao	datetime		Ngày tạo	
6	NgayCNCuoi	datetime		Ngày cập nhật cuối	
7	NguCanhSuDung	nvarchar(50)		Ngữ cảnh sử dụng	
8	NoiDungXml	ntext		Nội dung dưới dạng xml	
9	HinhAnh	nvarchar(50)		Đường dẫn tập tin hình ảnh	

Bảng PL - 27 Chi tiết bảng Thể hiện mẫu

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int	Khóa chính	Mã	
2	Ten	nvarchar(50)		Tên	
3	MoTa	nvarchar(50)		Mô tả	
4	Dll	nvarchar(50)		Đường dẫn tập tin dll	

Bảng PL - 28 Chi tiết bảng NgonNguPhatSinh

STT	Tên	Kiểu	Ràng buộc	Ý nghĩa	Ghi chú
1	Ma	Int	Khóa chính	Mã	
2	Ten	nvarchar(50)		Tên	
3	MoTa	nvarchar(50)		Mô tả	
4	Dll	nvarchar(50)		Đường dẫn tập tin dll	

Bảng PL - 29 Chi tiết bảng NgonNguNhapMoi

Thiết kế chỉ mục:

Một trong những yêu cầu đặt ra đối với hệ thống quản lý mẫu giao diện đó là hỗ trợ tìm kiếm nhanh trên các đối tượng mà hệ thống quản lý đó là thành phần mẫu, mẫu và thể hiện mẫu. Để giải quyết vấn đề này cần tạo chỉ mục trên những trường dữ liệu thích hợp. Với sơ đồ logic dữ liệu như trên, nhận thấy rằng một trong những cách tìm kiếm nhanh là thu hẹp phạm vi tìm kiếm bằng cách xác định các đối tượng như: thành phần cha đối với thành phần, mẫu cha đối với mẫu, và mẫu đối với thể hiện mẫu. Từ đó, ta chỉ phải tìm kiếm trong phạm vi những đối tượng có liên quan đến đối tượng được xác định trước.

STT	Tên	Loại	Tên bảng	Tên cột	Ý nghĩa
1	IX_ThanhPhan	CLUSTERED INDEX	Thanh Phan	MaCha	Sắp xếp bảng Thanh Phan theo MaCha
2	IX_Mau	CLUSTERED INDEX	Mau	MaCha	Sắp xếp bảng Mau theo MaCha
3	IX_TheHienMau	CLUSTERED INDEX	TheHienMau	MaMau	Sắp xếp bảng TheHienMau theo MaMau

Bảng PL - 30 Danh sách chỉ mục được thiết kế

Thiết kế các stored procedure:

Các thao tác giữa hệ thống quản lý mẫu và cơ sở dữ liệu tương đối nhiều và phức tạp không thể tiến hành dưới hình thức những câu truy vấn đơn giản và rời rạc. Chính vì thế tất cả các thao tác này sẽ được thiết kế thành các stored procedure để có thể kết hợp với việc tạo transaction, các mức cô lập, cơ chế khóa nhằm đảm bảo tính toàn vẹn và nhất quán của cơ sở dữ liệu.

STT	Tên	Bảng dữ liệu	Ý nghĩa
1	sp_ThemQuyền	Quyền	Thêm quyền mới
2	sp_CapNhatQuyền	Quyền	Cập nhật thông tin quyền
3	sp_XoaQuyền	Quyền	Xóa quyền, không được xóa khi có quyền đang được gán cho một người sử dụng nào đó.
4	sp_XoaDSQuyền	Quyền	Xóa toàn bộ quyền
5	sp_LayDSQuyền	Quyền	Lấy danh sách quyền
6	sp_ThemNguoiDung	NguoiDung	Thêm một người dùng, đồng thời gán quyền cho người dùng này
7	sp_LayNguoiDung	NguoiDung	Lấy thông tin người dùng theo mã
8	sp_CapNhat NguoiDung	NguoiDung	Cập nhật thông tin của người dùng
9	sp_LayDSNguoiDung	NguoiDung	Lấy danh sách người dùng
10	sp_LayDSNguoiDungTheoQuyền	NguoiDung	Lấy một nhóm người dùng có cùng quyền trên hệ thống
11	sp_XoaNguoiDung	NguoiDung	Xóa một người dùng, không cho sử dụng hệ thống nữa
12	sp_XoaDSNguoiDung	NguoiDung	Xóa toàn bộ danh sách người dùng
13	sp_ThemLoaiGiaoDien	LoaiGiaoDien	Thêm một loại giao diện mới
14	sp_XoaLoaiGiaoDien	LoaiGiaoDien	Xóa một loại giao diện
15	sp_CapNhat LoaiGiaoDien	LoaiGiaoDien	Cập nhật loại giao diện
16	sp_LayDS LoaiGiaoDien	LoaiGiaoDien	Lấy danh sách loại giao diện
17	sp_XoaDS LoaiGiaoDien	LoaiGiaoDien	Xóa toàn bộ các loại giao diện
18	sp_ThemLoaiMau	LoaiMau	Thêm một loại mẫu mới mà hệ thống cần quản lý
19	sp_XoaLoaiMau	LoaiMau	Xóa một loại mẫu. Chỉ được xóa khi nó không bị tham chiếu đến.
20	sp_CapNhatLoaiMau	LoaiMau	Cập nhật thông tin loại mẫu.
21	sp_XoaDSLoaiMau	LoaiMau	Xóa toàn bộ các loại mẫu
22	sp_LayDSLoaiMau	LoaiMau	Lấy toàn bộ danh sách loại

			mẫu
23	sp_ThemThanhPhan	ThanhPhan	Thêm một thành phần mới
24	sp_XoaThanhPhan	ThanhPhan	Xóa một thành phần. Chỉ được xóa khi không có thành phần nào có MaCha = Ma của thành phần
25	sp_CapNhatThanhPhan	ThanhPhan	Cập nhật thông tin của thành phần
26	sp_LayThanhPhan	ThanhPhan	Lấy một thành phần theo mã
27	sp_LayDSThanhPhan	ThanhPhan	Lấy danh sách các thành phần mà hệ thống có quản lý.
28	sp_XoaDSThanhPhan	ThanhPhan	Xóa danh sách thành phần
29	sp_ThemMau	Mau	Thêm một mẫu giao diện mới
30	sp_XoaMau	Mau	Xóa một mẫu giao diện. Chỉ được xóa khi nó không có mẫu nào có MaCha = Ma của nó.
31	sp_CapNhatMau	Mau	Cập nhật thông tin mẫu
32	sp_LayMau	Mau	Lấy thông tin một mẫu theo mã
33	sp_LayDSMau	Mau	Lấy danh sách các mẫu giao diện mà hệ thống quản lý
34	sp_LayDSMauTheoLoaiMau	Mau	Lấy danh sách mẫu theo loại
35	sp_XoaDSMau	Mau	Xóa toàn bộ danh sách mẫu
36	sp_ThemTheHienMau	TheHienMau	Thêm một thể hiện mẫu
37	sp_XoaTheHienMau	TheHienMau	Xóa thể hiện mẫu.
38	sp_CapNhatTheHienMau	TheHienMau	Cập nhật thông tin thể hiện mẫu
39	sp_LayTheHienMau	TheHienMau	Lấy một thể hiện mẫu theo mã
40	sp_LayDSTheHienMau	TheHienMau	Lấy danh sách thể hiện mẫu
41	sp_LayDSTheHieMauTheoMaMau	TheHienMau	Lấy nhóm thể hiện mẫu thuộc về cùng một mẫu
42	sp_XoaDSTheHienMau	TheHienMau	Xóa toàn bộ danh sách thể hiện mẫu
43	sp_ThemNgonNguPhatSinh	NgonNguPhatSinh	Thêm module phát sinh mã nguồn
44	sp_XoaNgonNguPhatSinh	NgonNguPhatSinh	Xóa module phát sinh mã nguồn.
45	sp_CapNhatNgonNguPhatSinh	NgonNguPhatSinh	Cập nhật thông tin module phát sinh mã nguồn
46	sp_LayNgonNguPhatSinh	NgonNguPhatSinh	Lấy thông tin một module phát sinh mã nguồn
47	sp_LayDS	NgonNgu	Lấy danh sách module phát

	NgonNguPhatSinh	PhatSinh	sinh mã nguồn
48	sp_XoaDS NgonNguPhatSinh	NgonNgu PhatSinh	Xóa danh sách module phát sinh mã nguồn
47	sp_Them NgonNguNhapMoi	NgonNgu NhapMoi	Thêm module phát sinh nhập vào mã nguồn
50	sp_Xoa NgonNguNhapMoi	NgonNgu NhapMoi	Xóa module nhập vào mã nguồn.
51	sp_CapNhat NgonNguNhapMoi	NgonNgu NhapMoi	Cập nhật thông tin module nhập vào mã nguồn
52	sp_Lay NgonNguNhapMoi	NgonNgu NhapMoi	Lấy thông tin một module nhập vào mã nguồn
53	sp_LayDS NgonNguNhapMoi	NgonNgu NhapMoi	Lấy danh sách module nhập vào mã nguồn
54	sp_XoaDS NgonNguNhapMoi	NgonNgu NhapMoi	Xóa danh sách module nhập vào mã nguồn

Bảng PL - 31 Danh sách các store procedure được thiết kế

Tài liệu tham khảo

- [1] Nguyễn Trần Minh Thư. Báo cáo Luận Văn Cao Học. Trường Đại học Khoa học Tự Nhiên. Năm 2004.
- [2] Scott W. Ambler, userInterfaceDesign, <http://www.ambysoft.com>.
- [3] Tidwell J., Common Ground: A Pattern Language for Human-Computer Interface Design (1999).
- [4] Laakso S.A, User Interfaces, Lecture notes of the 581391-1 User Interfaces course, series D405, Department of Computer Science, University of Helsinki, 2000.
- [5] Microsoft Corporation, MSDN Library.
- [6] CodeProject, www.codeproject.com.

KHOA CNTT