

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

**NGUYỄN MINH BÌNH - DIỆP HUỲNH ANH**

**TÌM HIỂU MÔ HÌNH VÀ CÔNG NGHỆ VỀ  
LUỒNG CÔNG VIỆC VÀ XÂY DỰNG ỨNG  
DỤNG VỀ TỔ CHỨC THI**

**KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT**

**TP. HCM, NĂM 2010**

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

**NGUYỄN MINH BÌNH – 0612023  
DIỆP HUỲNH ANH – 0612003**

**TÌM HIỂU MÔ HÌNH VÀ CÔNG NGHỆ VỀ  
LUỒNG CÔNG VIỆC VÀ XÂY DỰNG ỨNG  
DỤNG VỀ TỔ CHỨC THI**

**KHÓA LUẬN TỐT NGHIỆP CỦ NHÂN CNTT**

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

**KHÓA 2006 - 2010**

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TpHCM, ngày ..... tháng ..... năm .....  
Giáo viên hướng dẫn  
[Ký tên và ghi rõ họ tên]

## **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT.  
TpHCM, ngày ..... tháng ..... năm .....  
Giáo viên phản biện  
[Ký tên và ghi rõ họ tên]

## LỜI CẢM ƠN

Trước hết, chúng tôi xin gửi lời cảm ơn đến Quý Khoa Công nghệ Thông tin - Trường Đại Học Khoa Học Tự Nhiên - Đại Học Quốc Gia Thành phố Hồ Chí Minh đã tạo mọi điều kiện để chúng tôi có thể thực hiện khóa luận tốt nghiệp này.

Xin gửi lời cảm ơn sâu sắc đến PGS. TS. Đồng Thị Bích Thủy - giáo viên hướng dẫn của chúng tôi, đã luôn tận tình chỉ dạy và hướng dẫn trong suốt khoảng thời gian chúng tôi thực hiện khóa luận.

Đồng thời, chúng tôi cũng xin gửi lời cảm ơn chân thành đến các thầy cô giảng dạy trong suốt khoảng thời gian 4 năm đại học đã luôn tận tình chỉ bảo, truyền đạt kiến thức và kinh nghiệm quý báu cho chúng tôi; cảm ơn các bạn, những người đã cùng chúng tôi đi suốt chặng đường đại học với những vui buồn, sẻ chia trong việc học cũng như cuộc sống.

Cuối cùng, chúng tôi cũng xin gửi lời cảm ơn đến những người đã giúp đỡ chúng tôi trong suốt thời gian thực hiện khóa luận: Ông Scott Guthrie (Microsoft Product Manager) và Ông Rob Relyea (Microsoft .Net Developer); cùng tất cả những người đã luôn ở bên, ủng hộ và giúp đỡ chúng tôi hoàn thành bản luận văn này.

Tuy nhiên, dù đã cố gắng hết sức để thực hiện đề tài khóa luận, nhưng chúng tôi cũng không thể tránh khỏi những thiếu sót. Vì vậy, kính mong quý thầy cô và bạn đọc tận hình góp ý.

Nhóm sinh viên thực hiện:

Nguyễn Minh Bình – Diệp Huỳnh Anh

## ĐỀ CƯƠNG CHI TIẾT

<b>Tên Đề Tài:</b> Tìm hiểu mô hình và công nghệ về luồng công việc và xây dựng ứng dụng về tổ chức thi
<b>Giáo viên hướng dẫn:</b> PGS TS Đồng Thị Bích Thủy
<b>Thời gian thực hiện:</b> 14/12/2009 - 30/06/2010
<b>Sinh viên thực hiện:</b> Nguyễn Minh Bình - 0612023 Diệp Huỳnh Anh - 0612003
<b>Loại đề tài:</b> Tìm hiểu công nghệ, xây dựng ứng dụng

### Nội dung đề tài:

- Tóm tắt nội dung đề tài:**

Nội dung đề tài sẽ bao gồm:

- Khảo sát mô hình và ngôn ngữ đặc tả mô hình quản lý dòng công việc theo chuẩn của tổ chức WfMC.
- Khảo sát phần mềm Windows Workflow Foundation.
- Khảo sát một ứng dụng quản lý đào tạo thực tế, đặc biệt phân tích những tình huống có thể gây ra hiện tượng thất cố chai khi các thẻ hiện của những quy trình quản lý cùng diễn ra. Các quy trình cần được giám sát các luồng công việc là: Quy trình nhập điểm và kiểm tra điểm thi, quy trình xét cấp chứng chỉ/ văn bằng, quy trình in chứng chỉ/ văn bằng.
- Đặc tả mô hình dòng công việc của ứng dụng và cài đặt thử nghiệm với phần mềm mã nguồn mở đã chọn lọc, chú ý phát hiện tự động hiện tượng thất cố chai để cảnh báo những người khai thác.

- Các yêu cầu của đề tài**

- Thiết kế phần mềm theo kiến trúc 3 lớp

2. Chú ý xử lý hiện tượng thất cỗ chai.

- **Kết quả dự kiến**

Một ứng dụng theo dõi các luồng công việc của những quy trình quản lý đào tạo

- **Phương pháp thực hiện**

1. Phân chia công việc trong nhóm
2. Thảo luận với nhau và với giảng viên hướng dẫn
3. Báo cáo tiến độ hàng tuần trực tiếp cho đến giai đoạn lập trình

#### **Kế Hoạch Thực Hiện:**

Công việc	Bắt đầu	Kết thúc	Sinh viên thực hiện
Tìm hiểu tổng quan	16/01/2010	20/3/2010	
Tìm hiểu các bài báo về Hệ quản trị luồng công việc	16/01/2010	23/01/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Tìm hiểu về Tổ chức Thế Giới Workflow Management Coalition và các chuẩn mô hình hóa luồng công việc	23/01/2010	30/01/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Tìm hiểu các luận văn khóa trước về đề tài có liên quan đến luồng công việc	23/01/2010	30/01/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Tìm hiểu Windows Workflow Foundation	30/1/2010	10/03/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Tìm hiểu và xác định các yêu cầu chức năng của phần mềm quản lý quy trình thi, cấp chứng chỉ và in văn bằng	10/3/2010	20/03/2010	Nguyễn Minh Bình
Phân tích yêu cầu phần mềm	20/03/2010	16/04/2010	
Phân tích Nội dung Luồng công việc của quy trình thi, cấp	20/03/2010	05/04/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình

chứng chỉ và in văn bằng của các đơn vị đào tạo chứng chỉ quốc gia			
Phân tích yêu cầu phần mềm	05/04/2010	10/04/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Xây dựng mô hình dữ liệu mức quan niệm	10/04/2010	16/04/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Thiết kế phần mềm	16/04/2010	07/07/2010	
Thiết kế cơ sở dữ liệu phù hợp với quy trình thực tế; xây dựng mô hình dữ liệu mức vật lý	16/04/2010	20/04/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
Thiết kế các giao diện; xây dựng mô hình Usecase cho ứng dụng	16/04/2010	20/04/2010	Diệp Huỳnh Anh
Xây dựng các yêu cầu chức năng cho ứng dụng	20/04/2010	07/07/2010	Diệp Huỳnh Anh, Nguyễn Minh Bình
<b>Xác nhận của GVHD</b>	<b>Ngày 07 tháng 07 năm 2010</b> <b>SV Thực hiện</b>		
	<b>Diệp Huỳnh Anh      Nguyễn Minh Bình</b>		

●●●

## **Mục lục**

Chương 1: Đề Dẫn Đề Tài .....	25
1. Chủ đề.....	25
2. Yêu cầu của ứng dụng.....	26
3. Nội dung cần thực hiện để đáp ứng yêu cầu trên: .....	28
3.1. Tìm hiểu thực tế ứng dụng .....	28
3.2. Khảo sát và lựa chọn môi trường công nghệ phù hợp .....	28
3.2.1. Khảo sát: .....	28
3.2.2. Nhu cầu.....	29
3.2.3. Các công nghệ .....	29
3.2.4. Lựa chọn môi trường: .....	30
4. Bố cục khóa luận tốt nghiệp.....	31
Chương 2: Luồng công việc và ứng dụng luồng công việc .....	33
1. Tổng quan về mô hình luồng công việc .....	33
1.1. Luồng công việc .....	33
1.2. Mô hình luồng công việc.....	35
1.3. Sự thực thi các LCV .....	35
2. Tầm quan trọng trong việc đưa LCV và MH LCV vào ứng dụng .....	36
3. Mô hình hóa luồng công việc và vấn đề tồn tại .....	37
4. <i>Workflow Management Coalition (WfMC)</i> .....	38
4.1. Mô hình tham chiếu LCV ( <i>Workflow Reference Model - WfRM</i> ) .....	38
4.2. Các ngôn ngữ MHH LCV .....	43
4.2.1. <i>XML Process Definition Language (XPDL)</i> : .....	43
4.2.2. Wf-XML:.....	44
4.3. Các loại LCV.....	45
4.3.1. <i>Production</i> .....	45
4.3.2. Administrative .....	46
4.3.3. Collaborative .....	46

4.3.4. Ad-Hoc .....	47
5. Yêu cầu của một ứng dụng quản lý LCV tổng quát.....	47
Chương 3: Môi trường công nghệ để xây dựng một ứng dụng quản lý luồng công việc .....	49
1. <i>Workflow Management System</i> (viết tắt là WfMS): .....	49
2. <i>Windows Workflow Foundation</i> (viết tắt là WF) .....	50
2.1. Đặc điểm .....	50
2.2. Các loại LCV trong WF:.....	52
2.2.1. <i>Sequential Workflow</i> ( <i>Workflow tuần tự</i> ):.....	52
2.2.2. <i>State-Machine Workflow</i> ( <i>Workflow trạng thái</i> ):.....	53
2.2.3. <i>Rules-based Workflow</i> ( <i>Workflow dựa vào quy luật</i> ).....	53
2.3. Kiến trúc cơ bản của WF.....	54
2.3.1. Activity và Custom Activity.....	54
2.3.2. WF Runtime .....	55
2.4. Thiết kế LCV trong WF .....	56
2.4.1. C#, VB.net:.....	56
2.4.2. XAML:.....	57
2.5. Sự lưu trú của WF trên ứng dụng: .....	59
3. Sự khác nhau giữa WfMC và WF .....	60
3.1. Về ngôn ngữ MHH LCV.....	60
3.2. Về phân loại LCV .....	61
3.2.1. <i>Production Workflow</i> : .....	62
3.2.2. <i>Administrative Workflow</i> :.....	62
3.2.3. <i>Collaborative Workflow</i> : .....	62
3.2.4. <i>Ad-hoc</i> : .....	63
4. Windows Presentation Foundation (WPF).....	63
4.1. Tổng quan.....	63
4.2. XAML .....	64
4.3. Những tính năng nổi bật.....	64

4.3.1.	Data Binding (liên kết dữ liệu).....	64
4.3.2.	Layout.....	65
4.3.3.	Resource .....	65
4.3.4.	Style.....	65
4.3.5.	Transformation .....	66
4.3.6.	Animation.....	66
	Chương 4: Hiện thực ứng dụng.....	67
1.	Tổng quan về LCV tổ chức thi, cấp chứng chỉ quốc gia.....	67
1.1.	Các quy trình ứng dụng.....	68
1.1.1.	Quy trình tổ chức thi.....	68
1.1.2.	Quy trình cấp chứng chỉ .....	70
1.2.	Các vấn đề còn tồn tại .....	72
2.	Yêu cầu chức năng .....	73
2.1.	Chức năng thông báo.....	74
2.2.	Chức năng theo dõi công việc .....	74
2.2.1.	Đối với các nhân viên thực hiện các công việc: .....	74
2.2.2.	Đối với người quản lý: .....	75
2.3.	Chức năng cập nhật thông tin.....	75
2.4.	Chức năng thống kê.....	76
2.4.1.	Thống kê tổng thể: .....	76
2.4.2.	Thống kê chi tiết:.....	76
2.4.3.	Thống kê so sánh: .....	77
2.4.4.	Thống kê theo thời gian:.....	77
2.5.	Chức năng cảnh báo: .....	77
2.6.	Chức năng phân quyền:.....	78
2.7.	Chức năng biểu diễn luồng công việc dưới dạng sơ đồ trực quan.....	78
3.	Yêu cầu phi chức năng .....	79
3.1.	Tính tiến hóa.....	79
3.2.	Yêu cầu về giao diện .....	79

3.3.	Tính hiệu quả.....	80
4.	Phân tích mức quan niệm .....	80
4.1.	Mô hình dữ liệu ở mức quan niệm .....	80
4.1.1	Nhân viên .....	83
4.1.2	Nhân viên quản lý & Nhân viên thừa hành .....	84
4.1.3	Công việc .....	84
4.1.4	Phân công.....	85
4.1.5	Đợt thi .....	85
4.1.6	Tiến độ .....	86
4.1.7	Ghi chú.....	87
4.1.8	Chứng chỉ.....	87
4.2.	Mô hình xử lý mức quan niệm .....	87
4.2.1.	Tổng quan.....	87
4.2.2.	Quản lý nhân viên.....	90
4.2.3.	Quản lý quy trình.....	90
4.2.4.	Thống kê.....	91
4.2.5.	Thực thi quy trình.....	92
5.	Cài đặt mức vật lý .....	94
6.	Mô tả chức năng và các giao diện .....	95
6.1.	Thiết kế User-case .....	95
6.1.1.	Lược đồ User-case:.....	95
6.1.2.	Danh sách các Actor.....	96
6.1.3.	Danh sách các Use-case .....	96
6.2.	Thiết kế giao diện.....	97
6.2.1.	Cấu trúc các màn hình xử lý chính.....	97
6.2.2.	Danh sách các màn hình xử lý chính.....	98
7.	Giải pháp kĩ thuật .....	99
7.1.	Chức năng hiển thị luồng công việc dạng sơ đồ: .....	99
7.2.	Chức năng thống kê tình trạng các đợt thi đang tiến hành: .....	100

7.3. Custom Activity .....	101
Chương 5: Tổng kết .....	104
1. Tóm tắt công việc đã làm .....	104
2. Đặc điểm của Công việc .....	105
2.1. Nội dung .....	105
2.2. Hướng phát triển nhằm giải quyết những hạn chế về mặt nội dung.....	107
LỜI KẾT .....	109
Tài liệu tham khảo.....	110
PHỤ LỤC.....	111
1. XML Schema .....	111
2. Định dạng mẫu tập tin Designer trong WF .....	117
3. Các sản phẩm ứng dụng sử dụng XPDL .....	120
4. Mô tả chi tiết từng màn hình xử lý chính.....	124
4.1. Màn hình Đăng nhập:.....	124
4.2. Trang Báo lỗi: .....	125
4.3. Trang Nhân viên:.....	126
4.3.1. Màn hình chính.....	126
4.3.2. Thông tin cá nhân:.....	127
4.4. Trang Quản Lý: .....	127
4.4.1. Cập nhật đợt thi: .....	128
4.3.2. Thống kê Lược Đồ Gantt: .....	129
4.3.3. Cập nhật nhân viên:.....	130
4.3.4. Phân công theo đợt thi.....	130
4.3.5. Phân công theo kỳ .....	132
4.3.6. Thống kê chi tiết.....	133
4.3.7. Thống kê so sánh.....	134
4.3.8. Thống kê tổng hợp .....	135
4.3.9. Thêm tiêu chí thống kê.....	136
4.3.10. Thay đổi thời gian thực hiện công việc theo đợt thi .....	137

4.3.11. Thay đổi thời gian thực hiện công việc.....	138
4.3.12. Xem bản báo cáo kết quả công việc của nhân viên.....	138

## **Danh Mục Hình**

Hình 1. Luồng công việc Thi tuyển sinh đại học dưới góc nhìn của thí sinh .....	34
Hình 2. Luồng công việc Tổ chức tuyển sinh đại học dưới góc nhìn của đơn vị tổ chức tuyển sinh.....	35
Hình 3. WfRM .....	39
Hình 4. Mô hình chuyển đổi trạng thái .....	42
Hình 5. Các mô hình LCV tuần tự .....	53
Hình 6. MH LCV trạng thái .....	53
Hình 7. Cấu trúc luồng công việc .....	54
Hình 8. HelloWorld.....	59
Hình 9. Quy trình thi và cấp chứng chỉ.....	68
Hình 10. Quy trình tổ chức thi .....	69
Hình 11. Quy trình cấp chứng chỉ.....	72
Hình 12. Nhiều đợt thi có thể diễn ra song song.....	73
Hình 13. Biểu diễn quy trình dưới dạng sơ đồ Gantt .....	79
Hình 14. Mô hình dữ liệu quan niệm .....	81
Hình 15. Mô hình xử lý tổng quát.....	88
Hình 16. Mô hình xử lý Quản lý Nhân viên .....	90
Hình 17. Mô hình xử lý Quản lý Quy trình.....	91

Hình 18. Mô hình xử lý Thông kê.....	92
Hình 19. Mô hình xử lý Thực thi quy trình .....	93
Hình 20. Mô hình dữ liệu mức vật lý .....	95
Hình 21. Lược đồ Use-case.....	96
Hình 22. Cấu trúc màn hình chính .....	98
Hình 23. WorkItem .....	102
Hình 24. Màn hình đăng nhập.....	124
Hình 25. Trang báo lỗi .....	125
Hình 26. Màn hình chính nhân viên thừa hành.....	126
Hình 27. Màn hình cập nhật thông tin cá nhân .....	127
Hình 28. Thiết kế menu trang quản lý .....	127
Hình 29. Màn hình cập nhật đợt thi .....	128
Hình 30. Màn hình thống kê lượt đồ Gantt.....	129
Hình 31. Màn hình cập nhật nhân viên .....	130
Hình 32. Màn hình phân công theo đợt thi .....	131
Hình 33. Màn hình phân công theo kì.....	132
Hình 34. Màn hình thống kê chi tiết .....	133
Hình 35. Màn hình thống kê so sánh .....	134
Hình 36. Màn hình thống kê tổng hợp .....	135

Hình 37. Màn hình thêm tiêu chí thống kê .....	136
Hình 38. Màn hình thay đổi thời gian thực hiện công việc theo đợt thi .....	137
Hình 39. Màn hình thay đổi thời gian thực hiện công việc tổng thể.....	138
Hình 40. Màn hình xem báo cáo của nhân viên.....	138

## **Danh mục bảng**

Bảng 1. Các đối tượng dữ liệu được lưu trữ .....	81
Bảng 2. Ý nghĩa kí hiệu trong mô hình dữ liệu mức quan niệm.....	82
Bảng 3. Thông tin chi tiết thực thể nhân viên .....	84
Bảng 4.Thông tin chi tiết thực thể Nhân viên thừa hành .....	84
Bảng 5.Thông tin chi tiết thực thể công việc .....	85
Bảng 6.Thông tin chi tiết thực thể Phân Công.....	85
Bảng 7.Thông tin chi tiết thực thể Đợt thi .....	85
Bảng 8. Thông tin chi tiết thực thể tiến độ.....	86
Bảng 9. Thông tin chi tiết thực thể Ghi chú.....	86
Bảng 10. Thông tin chi tiết thực thể chứng chỉ.....	87
Bảng 11. Các dòng dữ liệu di chuyển trong mô hình xử lý tổng quát .....	89
Bảng 12. Các dòng dữ liệu di chuyển trong mô hình xử lý quản lý nhân viên.....	90
Bảng 13. Các dòng dữ liệu di chuyển trong mô hình xử lý thông kê .....	92
Bảng 14. Các dòng dữ liệu di chuyển trong mô hình xử lý thực thi quy trình .....	93
Bảng 15. Danh sách các Actor .....	96
Bảng 16. Danh sách các Use-case.....	97
Bảng 17. Danh sách các màn hình xử lý chính .....	99
Bảng 18. Danh sách control màn hình đăng nhập.....	125

Bảng 19. Danh sách control màn hình chính của nhân viên .....	126
Bảng 20. Danh sách control màn hình cập nhật đợt thi .....	128
Bảng 21. Danh sách control màn hình xử lý cập nhật đợt thi.....	128
Bảng 22. Danh sách control màn hình thêm đợt thi.....	129
Bảng 23. Danh sách control màn hình phân công theo đợt thi .....	131
Bảng 24. Danh sách control màn hình phân công theo kì .....	132
Bảng 25. Danh sách control màn hình thông kê chi tiết .....	133
Bảng 26. Danh sách control màn hình thống kê so sánh .....	134
Bảng 27. Danh sách control màn hình thống kê tổng hợp .....	135
Bảng 28. Danh sách control màn hình thêm tiêu chí thống kê .....	136
Bảng 29. Danh sách control Màn hình thay đổi thời gian thực hiện công việc theo đợt thi .....	137
Bảng 30. Danh sách control Màn hình xem báo cáo kết quả công việc .....	138

## LỜI NÓI ĐẦU

Công nghệ Thông tin đang ngày càng phát triển và chiếm vị trí quan trọng trong nhiều lĩnh vực như giáo dục, quốc phòng, sản xuất, thương mại v.v... Với khả năng đáp ứng nhiều yêu cầu khác nhau của các ngành công nghiệp, đồng thời với sự bùng nổ của Internet và Công nghệ Phần mềm, Công nghệ Thông tin đang ngày càng được ứng dụng rộng rãi. Ngày nay, các doanh nghiệp đang dần tin học hóa một phần hoặc toàn bộ các thành phần, các hoạt động để có thể giải phóng tối đa tài nguyên lao động, tăng cường tài nguyên chất xám. Cũng nhờ đó mà chất lượng và năng suất công việc cũng tăng vọt, một người có thể đảm đương nhiều vai trò và công việc hơn. Không những thế, Công nghệ Thông tin còn giúp các doanh nghiệp có thể gắn kết với nhau và trao đổi thông tin dễ dàng hơn, mở rộng thị trường một cách nhanh chóng và hiệu quả, linh hoạt trong các hoạt động quản lý, khai thác tốt hơn các nguồn thông tin, v.v...

Từ các nghiệp vụ chuyên môn cho đến các nghiệp vụ quản lý, Công nghệ Phần mềm đã trợ giúp nhiều doanh nghiệp một cách đắc lực và ngày càng mạnh mẽ hơn. Người thư ký không cần phải sử dụng đến những máy đánh chữ mà cứ mỗi lần sai lại phải bỏ đi một tờ giấy và làm lại từ đầu. Các phần mềm soạn thảo và định dạng văn bản giúp họ có thể tạo ra văn bản đẹp hơn, trình bày bắt mắt hơn, khi gặp lỗi có thể sửa ngay trên văn bản; đồng thời có thể in, lưu trữ, bảo mật văn bản v.v... Đặc biệt, với người quản lý, dưới sự trợ giúp của Công nghệ Phần mềm, giờ đây không cần phải đi khắp nơi, tốn rất nhiều thời gian để điều khiển hoạt động và kiểm soát các công việc trong doanh nghiệp. Chỉ cần ngồi dưới máy vi tính, tất cả các thông tin cần thiết sẽ được cập nhật tự động; những bảng thống kê được máy tính lập trong nháy mắt, người quản lý có thể nắm bắt hoạt động của doanh nghiệp và nhanh chóng kiểm soát, đưa ra giải pháp kịp thời khi có sự cố.

Hơn nữa, với sự phát triển ngày càng nhanh của Công nghệ Thông tin, các phần mềm trợ giúp quản lý ngày càng được cải thiện. Từ những phần mềm được xây

dựng theo những phương thức cơ sở, ngày nay, con người đã có thể xây dựng các phần mềm quản lý theo dõi các quy trình nghiệp vụ của doanh nghiệp, theo nhu cầu riêng của từng công ty. Đó chính là những phần mềm quản lý luồng công việc. Những sản phẩm này có khả năng mô hình hóa luồng công việc, tạo ra sự nhất quán và tăng hiệu quả công việc cho riêng từng quy trình cụ thể. Những phần mềm thuộc loại này thỏa mãn được nhiều nhu cầu của doanh nghiệp nên ngày càng phát triển mạnh hơn và theo nhiều hướng khác nhau. Các chuẩn cho sự quản lý theo quy trình được tạo ra, theo đó, các nền tảng công nghệ đáp ứng các chuẩn này cũng được ra đời.

Chỉ nói riêng tại Việt Nam, việc theo dõi các nghiệp vụ xử lý công văn đơn từ trong các tổ chức nhà nước; công tác an ninh quốc phòng; theo dõi công tác tuyển sinh, thi cấp chứng chỉ trong các tổ chức giáo dục; theo dõi quá trình thực hiện các công việc thuộc quy trình nghiệp vụ trong các tổ chức doanh nghiệp,... là thực sự quan trọng và hầu như không thể cho phép xảy ra bất cứ sai sót nào. Công tác quản lý đối với các lĩnh vực này thật sự là một công việc khó khăn và nhạy cảm với các rủi ro. Tuy nhiên, hoạt động của công việc quản lý từ trước đến nay đa số đều được thực hiện thủ công, người quản lý phải tiếp xúc từng nhân viên (gặp trực tiếp, qua mạng liên lạc, qua hồ sơ...) mới có thể lấy đầy đủ thông tin, và phải tự tính toán thống kê thông qua một số chương trình nhỏ hỗ trợ tính toán... Những công việc này vốn tốn rất nhiều thời gian và công sức. Vì thế, việc xây dựng một chương trình hoàn thiện theo xu hướng phát triển chung của thế giới, hỗ trợ tối đa cho người quản lý là một trong những nhu cầu cần được giải quyết.

Chúng tôi chọn đề tài Xây dựng phần mềm luồng công việc cho đồ án tốt nghiệp trước hết vì hiện tại, ở Việt Nam hiện nay, những phần mềm xây dựng theo hướng này chưa nhiều, và thường tốn rất nhiều chi phí trong khi nhu cầu sở hữu ứng dụng như trên đối với doanh nghiệp cũng ngày càng tăng. Đặc biệt khi phần mềm luồng công việc này có thể giải quyết bài toán quy trình doanh nghiệp một cách hiệu quả, vì nó cho phép bạn mô hình hóa một cách trực quan và rõ ràng một quy trình của

doanh nghiệp. Hơn thế nữa, nó còn cho phép bạn theo dõi được hoạt động của quy trình khi thực thi và có thể thay đổi nó khi đang chạy.

Tuy nhiên, trong xây dựng phần mềm luồng công việc, có rất nhiều hướng đi, nhiều chuẩn, và nền tảng công nghệ khác nhau. Sau khi tìm hiểu và lựa chọn, chúng tôi quyết định sử dụng Windows Workflow Foundation (một thành phần trong nền tảng .Net 3.0) để xây dựng. Đồng thời, chúng tôi cũng quyết định xây dựng một phần mềm ứng dụng thực tế nhằm quản lý quy trình nghiệp vụ trong tổ chức thi, chấm thi, và cấp chứng chỉ quốc gia cho thí sinh dựa trên nền tảng đã chọn, chứ không xây dựng các phần mềm nhỏ minh họa. Vì vậy nội dung chính của luận văn sẽ bàn về nội dung của phần mềm và các hướng giải quyết.

## **Chương 1: Đề Dẫn Đề Tài**

*Chương này giới thiệu tóm lược về nội dung của khóa luận tốt nghiệp. Thay vì tập trung diễn giải các vấn đề lý thuyết càn thiêt, chúng tôi quyết định sử dụng chương mở đầu này để giới thiệu chung về sự phát triển chung của công nghệ thông tin và ứng dụng của nó vào các ngành công nghiệp khác nhau. Sau đó chúng tôi sẽ bàn đến yêu cầu chung của ứng dụng mà chúng tôi sẽ thiết kế trong khóa luận tốt nghiệp này, đồng thời giới thiệu bối cảnh của luận văn.*

### **1. Chủ đề**

Hiện nay, ứng dụng của công nghệ thông tin, đặc biệt là công nghệ phần mềm và hệ thống thông tin vào các nghiệp vụ kinh tế ngày càng phát triển sâu và rộng. Các doanh nghiệp càng mở rộng thì càng có nhu cầu sử dụng công nghệ thông tin vào trong các nghiệp vụ kinh doanh của mình. Vì thế, công nghệ thông tin không chỉ tác động đến các ngành công nghiệp ở các nghiệp vụ khác nhau mà còn tác động đến cả quy trình kinh doanh và quản lý của doanh nghiệp.

Trong các ngành công nghiệp khác nhau, công nghệ thông tin luôn thể hiện vai trò của mình. Các phần mềm hỗ trợ được tạo ra giúp các nhân viên của doanh nghiệp có thể thực hiện công việc của mình hiệu quả hơn, đồng thời có sự nhất quán giữa các công việc với nhau, nghĩa là, công nghệ thông tin không chỉ giúp công việc được thực hiện nhanh, tốt hơn mà còn giúp điều khiển các công việc giống nhau theo một quy trình, không rời rạc hay trùng lắp thông tin.

Tuy nhiên, bản chất mọi công việc trong doanh nghiệp là có liên quan chặt chẽ với nhau, nghĩa là, từng công việc tuy khác nhau, riêng rẽ nhưng chúng là một công đoạn trong một nghiệp vụ cụ thể. Từng công việc nhỏ được thực hiện theo một trình tự nhất định, gọi là quy trình, nhằm mục tiêu hoàn thành nghiệp vụ cụ thể. Vì thế, công nghệ phần mềm không chỉ có khả năng hỗ trợ những công việc nhỏ đó, mà hơn nữa, theo nhu cầu ngày càng cao của con người, công nghệ thông tin còn phát

triển mạnh hơn nữa, tạo ra các ứng dụng hỗ trợ cả quy trình nghiệp vụ của doanh nghiệp.

Hiện nay thế giới đã xuất hiện rất nhiều phần mềm xây dựng nhằm giúp đỡ tối đa các doanh nghiệp vận hành nghiệp vụ của họ theo đúng quy trình, đồng thời có cái nhìn tổng quan, cụ thể về sự vận hành đó nhằm quản lý, theo dõi và cập nhật tình trạng vận hành. Đó là phần mềm mô hình hóa luồng công việc. Phần mềm có khả năng giúp người sử dụng thực hiện các tác vụ của mình dễ dàng hơn, hoặc thể hiện kết quả của công việc lên chương trình nhằm giúp ứng dụng thống kê, theo dõi, và giám sát, điều khiển hoạt động của luồng công việc. Hơn nữa, những thống kê này có khả năng thể hiện lại (dưới nhiều dạng khác nhau) cho các nhà quản lý, giúp họ theo dõi tình trạng công việc và đưa ra các hướng xử lý nhanh chóng và kịp thời khi gặp sự cố.

Ở Việt Nam, các phần mềm xây dựng giúp hỗ trợ các công việc khác nhau cho doanh nghiệp đã phát triển rát rộng rãi. Tuy nhiên, những phần mềm quản lý dạng luồng công việc thì còn rất ít và đắt tiền, thường phải mua với giá cao từ các nước khác trên thế giới. Vì vậy, lựa chọn đề tài này cho khóa luận tốt nghiệp, chúng tôi hi vọng có thể tạo ra một phần mềm miễn phí có khả năng hỗ trợ tối đa trong các quy trình nghiệp vụ; ở đây là nghiệp vụ quản lý đào tạo, nhằm mục đích hỗ trợ người dùng ở cấp độ quản lý quy trình đào tạo, cụ thể là quy trình thi, cấp chứng chỉ và in văn bằng cho các đơn vị đào tạo chứng chỉ quốc gia các loại. Vì vậy, mục đích của ứng dụng được tạo ra là giúp xây dựng một ứng dụng quản lý luồng công việc liên quan đến quy trình thi, cấp chứng chỉ và in văn bằng dưới dạng sản phẩm phần mềm mô hình hóa luồng công việc, ứng dụng nền tảng công nghệ đã chọn, giúp người quản lý có cái nhìn tổng quát, và chi tiết vào từng tác vụ trên luồng công việc.

## 2. Yêu cầu của ứng dụng

Trước hết, ứng dụng được xây dựng cần đảm bảo đáp ứng yêu cầu mô hình hóa luồng công việc, nghĩa là:

- Các công việc trong luồng công việc được thực hiện theo đúng quy trình, đúng tác vụ, đúng người đã được phân công và trong thời hạn quy định.
- Có khả năng cập nhật tình trạng công việc dựa trên dữ liệu đầu vào được cung cấp từ người thực hiện công việc đó.

Ngoài ra, để hỗ trợ tối đa cho người quản lý, nhằm mục đích hỗ trợ người quản lý trong công việc của mình, ứng dụng cần có khả năng:

- Biểu diễn luồng việc dưới dạng sơ đồ luồng công việc, sơ đồ Gantt giúp người quản lý có cái nhìn trực quan trên tổng thể các quy trình công việc đang được thực thi.
- Thể hiện kết quả từng tác vụ cụ thể, rõ ràng.
- Thông báo, cảnh báo đối với những trường hợp có thể gây ra các rủi ro.
- Thống kê, so sánh các kết quả thực hiện từng thể hiện luồng công việc<sup>1</sup> (*workflow instance*) dưới nhiều dạng khác nhau đi từ tổng thể đến chi tiết.

Cuối cùng, ứng dụng phải đảm bảo các yêu cầu cơ bản của một phần mềm ứng dụng Thông tin, cũng như đáp ứng xu hướng Công nghệ Phần mềm hiện tại và tương lai:

- Giao diện đẹp, thân thiện, dễ sử dụng
- Không cần tương tác nhiều, không rườm rà, dư thừa.
- Ứng dụng Thông tin cần đảm bảo dữ liệu đủ, không sót thông tin, xử lý được hiện tượng thắt cổ chai, bảo mật tài khoản người dùng...

---

<sup>1</sup> Từ đây trở đi, chúng tôi sẽ dùng cụm từ "luồng công việc" để chỉ một thể hiện của một luồng công việc, và "mô hình luồng công việc" để chỉ phần mô tả cấu thành của một quy trình nghiệp vụ.

### **3. Nội dung cần thực hiện để đáp ứng yêu cầu trên:**

#### **3.1. Tìm hiểu thực tế ứng dụng**

Nội dung của ứng dụng mà chúng tôi sẽ thực hiện trong luận văn này, như đã đề cập trước đó, là sẽ thiết kế một ứng dụng thực tế hỗ trợ công việc quản lý quy trình thi, cấp chứng chỉ và in văn bằng các đơn vị đào tạo chứng chỉ quốc gia.

Theo như đã tìm hiểu, chúng tôi nhận thấy hiện nay, các đơn vị đào tạo chứng chỉ quốc gia thường thực hiện các công việc trong quy trình thi, cấp chứng chỉ và in văn bằng theo một luồng công việc cụ thể; nghĩa là trình tự cho các công việc cần thực hiện trong quy trình đã được định nghĩa và thực thi chính xác. Tuy nhiên, tất cả các công đoạn thực thi đều được báo cáo lại và kiểm tra một cách thủ công, mất rất nhiều thời gian của người quản lý. Chưa có một chương trình quản lý nào được xây dựng để theo dõi, kiểm soát và thống kê kết quả cũng như quá trình thực hiện của luồng công việc, khiến người quản lý phải tự kiểm tra, thống kê và ghi lại kết quả bằng tay. Hơn nữa, như chúng tôi đã đề cập trước đó (Mục Lời Mở Đầu), tại Việt Nam hiện nay, các ứng dụng hỗ trợ công việc chuyên môn của các đối tượng khác nhau phát triển rất nhanh, có rất nhiều phần mềm, đa dạng và phong phú. Tuy nhiên, một chương trình xây dựng ở mức luồng công việc, vận hành theo cơ chế của quy trình nghiệp vụ vẫn còn khiêm tốn, thường phải mua từ nước ngoài với giá rất đắt. Vì thế, nhu cầu tạo ra một chương trình hỗ trợ quản lý việc thi, cấp chứng chỉ và in văn bằng cho các trung tâm là điều cấp thiết và cần được triển khai.

#### **3.2. Khảo sát và lựa chọn môi trường công nghệ phù hợp**

##### **3.2.1. Khảo sát:**

Thực tế hiện nay đa số các đơn vị đào tạo chứng chỉ cấp quốc gia chưa sử dụng các ứng dụng đa tác vụ nhằm hỗ trợ các nhà quản lý trong việc thực hiện công việc của mình. Vì thế, mọi công việc của người quản lý đều thực hiện bằng tay, thông qua một số thiết bị hỗ trợ (điện thoại, tin nhắn sms, trình soạn thảo văn bản, máy

tính...), mất nhiều thời gian và rất khó cập nhật, thống kê cũng như theo dõi quá trình thực hiện.

### **3.2.2. Nhu cầu**

Xây dựng một ứng dụng hỗ trợ người quản lý thực hiện công việc của mình. Chương trình cần đáp ứng đầy đủ các yêu cầu (đã đề cập ở mục 2 chương 1). Như vậy, ứng dụng cần được xây dựng có khả năng đáp ứng những chức năng phức tạp, đồng thời giao diện đồ họa phải có khả năng thể hiện được luồng công việc dưới dạng sơ đồ, biểu đồ cũng như có khả năng biểu diễn được quá trình hiện tại trên thực tế một cách trực quan, dễ hiểu và gần gũi với người sử dụng.... Ngoài ra, các công việc trong quy trình được thực hiện có các ràng buộc về thời gian và trình tự nhất định, đòi hỏi ứng dụng phải xử lý các công việc đúng với luồng công việc đã được định nghĩa..

### **3.2.3. Các công nghệ**

- Xét về công nghệ hỗ trợ mô hình hóa luồng công việc: hiện nay, trên thế giới có rất nhiều kiến trúc (*framework*), ngôn ngữ, cũng như các chuẩn mô hình hóa khác nhau đã được đưa ra nhằm giải quyết bài toán luồng công việc. Chẳng hạn như các kiến trúc: ARIS, CIMOSA, DoDAF...; các ngôn ngữ mô hình hóa: Wf-XML, XPDL, BPMN...; các chuẩn mô hình hóa: WfMC, OASIS...<sup>2</sup>; các nền tảng hỗ trợ xây dựng phần mềm mô hình hóa luồng công việc: Các nền tảng hỗ trợ cho các phần mềm mã nguồn mở (Java), Windows Workflow Foundation hỗ trợ trên nền tảng .Net của Microsoft,...

- Về công nghệ thiết kế và xây dựng phần mềm có giao diện đồ họa cấp cao hiện nay cũng rất đa dạng. Với mỗi ngôn ngữ lập trình khác nhau, có các công nghệ khác nhau như:

---

<sup>2</sup> Xem chi tiết ở khóa luận tốt nghiệp "Tìm hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quy trình nghiệp vụ" , Khoa CNTT, trường ĐHKHTN tp HCM, năm 2008 - Chương 2

- Flash, Flex của Adobe, sử dụng ngôn ngữ Action Script
- Silverlight, Windows Presentation Foundation của Microsoft, sử dụng ngôn ngữ C# hoặc VB.Net
- ...

### **3.2.4. Lựa chọn môi trường:**

Trên thực tế, việc xây dựng ứng dụng là nhằm giải quyết các vấn đề kinh doanh thực tế. Những vấn đề này về bản chất rất khác nhau, cả về loại cũng như độ phức tạp của nó. Tuy nhiên, dù độ phức tạp của nó đến thế nào đi chăng nữa, hầu hết chúng ta đều giải quyết chúng theo cùng một cách: Trước hết, chúng ta cần chia nhỏ các vấn đề thành những phần nhỏ hơn, và nhỏ hơn nữa... (vấn đề thi, cấp chứng chỉ có thể xem là một ví dụ, chúng được chia ra làm nhiều tác vụ con); cho đến khi chúng có thể thực hiện cũng như quản lý dễ dàng. Sau đó, chúng ta cần xác định rõ trình tự cần thực hiện giữa các phần để có thể giải quyết được vấn đề. Các tác vụ nhỏ được chia cùng với trình tự thực hiện chúng tạo thành một chuỗi tác vụ độc lập mà mục đích của nó chỉ có thể đạt được khi chúng được thực thi đúng theo trình tự đã lập ra.

Các giải quyết bài toán nghiệp vụ đối với quy trình thi, cấp chứng chỉ và in bǎn bằng cho các đơn vị đào tạo chúng chỉ quốc gia được nghiên cứu trong khóa luận này cũng không nằm ngoài hướng nêu trên. Vì vậy, chúng tôi đã quyết định sẽ giải quyết bài toán theo hướng xây dựng phần mềm mô hình hóa luồng công việc.

Tuy nhiên, theo tìm hiểu các công nghệ Luồng Công Việc, có rất nhiều nền tảng đã được xây dựng nên, và với mỗi nền tảng, cũng có rất nhiều Engine. Mỗi Engine được xây dựng hỗ trợ xây dựng sản phẩm trên các môi trường khác nhau. Vì thế, sau khi cân nhắc, chúng tôi đã quyết định sử dụng Windows Workflow Foundation làm nền tảng xây dựng chương trình với lý do:

- Đây là nền tảng được Microsoft xây dựng, là một thành phần của .Net Framework nên dễ lập trình, dễ cài đặt hơn
- Đây là một công nghệ còn khá mới hiện nay, nhưng có khả năng phát triển cao trong tương lai
- Hướng xây dựng với Windows Workflow Foundation có thể triển khai, mở rộng và phát triển lâu dài, đồng thời cũng dễ dàng nâng cấp, dễ dàng tái sử dụng

Vì vậy, nội dung của khóa luận này sẽ giải quyết bài toán quản lý Quy Trình Thi, chấm thi và Cấp chứng chỉ trong nghiệp vụ quản lý đào tạo theo hướng sử dụng Windows Workflow Foundation làm nền tảng công nghệ. Vì Windows Workflow Foundation được viết bởi Microsoft nên đồng thời chúng tôi cũng quyết định sẽ sử dụng công nghệ mới Windows Presentation Foundation (cung cấp cùng với Windows Workflow Foundation trong bộ .Net) để xây dựng chương trình, đáp ứng các yêu cầu của ứng dụng.

#### **4. Bố cục khóa luận tốt nghiệp**

Nội dung của khóa luận sẽ bao gồm 5 chương:

**Chương 1: Dẫn dắt đề tài khóa luận tốt nghiệp.** Nội dung của chương sẽ hướng về yêu cầu chung của đề tài, phân tích khảo sát thực tế và đưa ra quyết định lựa chọn môi trường công nghệ nghiên cứu cũng như xác định hướng giải quyết cho phần mềm. Phần cuối của chương sẽ đề cập đến bố cục của luận văn.

**Chương 2: Luồng công việc và ứng dụng luồng công việc.** Nội dung chương này mô tả các kiến thức lý thuyết cơ bản về luồng công việc cũng như ứng dụng luồng công việc. Phần đầu của chương sẽ tìm hiểu về luồng công việc, mô hình hóa luồng công việc cũng như tầm quan trọng và các vấn đề tồn tại trong việc mô hình hóa luồng công việc và xây dựng ứng dụng hỗ trợ. Phần sau của chương sẽ đề cập

đến tổ chức WfMC cùng những thành quả của tổ chức này, đồng thời giới thiệu các chức năng cơ bản của một ứng dụng quản lý luồng công việc tổng quát.

**Chương 3: Môi trường công nghệ để xây dựng một ứng dụng quản lý luồng công việc.** Chương này trình bày kết quả tìm hiểu lý thuyết về các môi trường công nghệ phù hợp. Vì nội dung của chương có rất nhiều chi tiết đã được trình bày ở các luận văn trước, nên chúng tôi quyết định sẽ chỉ trình bày lại một phần kiến thức chung và tập trung vào những nội dung mới. Vì thế, phần đầu của chương sẽ giới thiệu sơ về WF (Windows Workflow Foundation), đồng thời phân tích những điểm khác nhau giữa các chuẩn do WfMC đưa ra và các chuẩn được WF sử dụng. Sau đó chúng tôi sẽ trình bày về WPF, nền tảng công nghệ được chúng tôi sử dụng chính trong thiết kế giao diện của ứng dụng.

**Chương 4: Hiện thực ứng dụng.** Chương này trình bày lại quá trình khảo sát, phân tích yêu cầu của ứng dụng, đồng thời trình bày kết quả thiết kế của chúng tôi. Phần đầu của chương là kết quả khảo sát luồng công việc tổ chức thi và cấp chứng chỉ quốc gia và yêu cầu cụ thể của ứng dụng quản lý luồng công việc loại này. Phần sau sẽ trình bày về kết quả phân tích ở cấp độ dữ liệu và kết quả thực tế mà chúng tôi đã thiết kế. Cuối chương, chúng tôi sẽ trình bày về các giải pháp thuật toán cũng như các phương pháp kỹ thuật được sử dụng để giải quyết các vấn đề phức tạp trong chương trình.

**Chương 5: Kết luận.** Đây là chương cuối cùng của luận văn. Chương này sẽ tổng kết lại các phần của chương trước. Đồng thời đưa ra những đặc điểm nổi bật của luận văn (bao gồm các ưu điểm và hạn chế). Cuối cùng, chúng tôi sẽ đưa ra một số hướng phát triển có thể nhằm giải quyết các hạn chế về mặt nội dung của ứng dụng.

## **Chương 2: Luồng công việc và ứng dụng luồng công việc**

*Phần mở đầu của khóa luận này đã bàn đến sự phát triển mạnh mẽ cũng như sự ứng dụng ngày một rộng rãi trong hầu hết các lĩnh vực của Công nghệ Thông tin, kèm theo đó là nhu cầu mỗi lúc một cấp thiết hơn đối với các ứng dụng quản lý đa năng, linh động và thật sự mạnh mẽ. Đó là các yêu cầu về các mức độ quản lý từ tổng quát đến chi tiết, về phương thức quản lý trực tiếp đến quản lý từ xa, về khả năng đưa ra các thông tin quan trọng hỗ trợ tối đa quá trình dự báo, cảnh báo cũng như giải quyết các rủi ro v.v... Và quan trọng hơn hết, một ứng dụng quản lý cấp cao phải thật sự linh động đối với sự thay đổi quy trình nghiệp vụ bên trong...*

*Phần này tiếp tục giới thiệu các khái niệm liên quan đến mô hình luồng công việc. Tiếp sau đó chúng tôi sẽ đề cập đến tầm quan trọng của một ứng dụng quản lý luồng công việc, cùng với các vấn đề tồn tại, và sự ra đời của tổ chức Workflow Management Coalition với các đóng góp của tổ chức này trong việc tiêu chuẩn hóa việc Mô Hình Hóa Luồng Công Việc. Phần cuối sẽ trình bày về các yêu cầu đối với một ứng dụng quản lý luồng công việc tổng quát.*

### **1. Tổng quan về mô hình luồng công việc**

#### **1.1. Luồng công việc**

Vào những năm tám mươi của thế kỷ hai mươi, khái niệm Luồng công việc (Workflow) mới được sử dụng lần đầu tiên trong ngành công nghệ phần mềm<sup>3</sup>. Tuy thế, sự xuất hiện của Luồng công việc và Mô hình luồng công việc trong cuộc sống của con người hẳn đã từ rất lâu. Giữa vô vàn các hoạt động của mình trong cuộc sống, có nhiều công việc mà con người lặp đi lặp lại nhiều lần. Một hoặc một số các công việc có thể được lặp lại theo chu kỳ nhiều giờ, nhiều ngày, nhiều tháng v.v... hoặc xảy ra bất kỳ lúc nào. Dựa vào kinh nghiệm đúc kết được, con người vạch ra các bước phải làm và dần dần hoàn thiện chúng cho các lần thực hiện sau. Nói khác

---

<sup>3</sup> <http://en.wikipedia.org/wiki/Workflow>

hơn, dựa vào kinh nghiệm và sự hiểu biết của mình, con người có thể vạch ra các kế hoạch; xác định thành phần con người, cũng như công cụ cần có để thực hiện công việc nhằm đạt được mục đích nhất định. Các bước cần thực hiện trong các kế hoạch đó có mối quan hệ với nhau và tạo thành luồng công việc.

*Luồng công việc là một tập có thứ tự các công việc tuân theo các quy tắc nhất định, được định sẵn sao cho việc thực hiện theo đó sẽ đạt được các mục đích công việc đã đề ra.* [4]

Khái niệm về luồng công việc là một khái niệm tương đối. Cùng tham gia vào một hệ thống trong cùng một môi trường, nhưng những đối tượng với vai trò khác nhau sẽ có các luồng công việc khác nhau. Hình vẽ dưới đây là ví dụ về các luồng công việc trong hệ thống tổ chức thi tuyển sinh đại học tại Việt Nam. Dưới góc nhìn của các thí sinh, luồng công việc thi tuyển sinh gồm 3 việc chính phải làm. Thực hiện luồng công việc này, các thí sinh đạt được mục đích là dự thi đại học thành công. Dưới góc nhìn của đơn vị tổ chức tuyển sinh thì lại khác, luồng công việc tổ chức thi tuyển sinh gồm có 9 công việc chính phải làm, thực hiện luồng công việc này sẽ đạt được mục đích là tổ chức thành công một kỳ tuyển sinh đại học.



**Hình 3. Luồng công việc Thi tuyển sinh đại học dưới góc nhìn của thí sinh**

---

<sup>4</sup> SHI Meilin, YANG Guangxin, XIANG Yong, WU Shangguang, Workflow Management Systems: A Survey, Department of Computer Science, Tsinghua University, Beijing, P.R.China 100084, 1998



**Hình 4. Luồng công việc Tổ chức tuyển sinh đại học dưới góc nhìn của đơn vị tổ chức tuyển sinh**

### 1.2. Mô hình luồng công việc

*Mô hình luồng công việc là một dạng biểu diễn luồng công việc thành sơ đồ.*<sup>5</sup>

Việc biểu diễn một LCV thành mô hình LCV gọi là *Mô hình hóa luồng công việc*. Một công việc thực tế sẽ mô hình hóa thành một thành phần xử lý của MH LCV, và quy tắc thực hiện công việc trong thực tế sẽ được mô hình hóa thành các quan hệ logic giữa các thành phần xử lý đó.

### 1.3. Sự thực thi các LCV

Üng với một quá trình thực thi LCV là một thể hiện của LCV đó. Một LCV có thể có nhiều thể hiện khác nhau do quá trình thực thi khác nhau. Mỗi LCV khi thực thi sẽ phải tuân thủ các quy tắc nhất định. Đó là các quy định về quy trình nghiệp vụ thực tế của tổ chức, doanh nghiệp tạo ra và thực thi LCV đó như các ràng buộc về sự tương tác với người thực hiện công việc, ràng buộc về thời gian thực thi, các điều kiện cho phép xảy ra sự kiện chuyển đổi trạng thái v.v... Ngoài ra, từng công việc thành phần bên trong một LCV cũng có các ràng buộc riêng của nó trong quá trình thực thi.

Ví dụ, công việc “Nhận đăng ký thi” trong ví dụ về LCV Tổ chức tuyển sinh đại học ở phần trước có thể có một số ràng buộc như: bắt đầu từ ngày  $t_1$  và kết thúc vào ngày  $t_2$ ; chỉ cho phép đăng ký khi các giấy tờ đều hợp lệ, thí sinh đăng ký phải tốt nghiệp trung học phổ thông trước đó, v.v...

---

<sup>5</sup> Từ đây về sau, chúng tôi dùng từ viết tắt LCV để thay cho cụm từ “Luồng Công việc” và MH LCV để thay cho cụm từ “Mô hình luồng công việc”.

## 2. Tầm quan trọng trong việc đưa LCV và MH LCV vào ứng dụng

Hiện nay, trên thế giới, đã có rất nhiều ứng dụng hỗ trợ thực thi và quản lý LCV. Những ứng dụng loại này được xem là các Hệ Quản Trị Luồng Công Việc (HQT LCV)<sup>6</sup>. Các HQT LCV ngày càng được sử dụng rộng rãi trong doanh nghiệp và ngày càng trở nên quan trọng hơn bởi:

- Các ứng dụng thường chỉ hỗ trợ người dùng thực hiện đúng công việc của mình. Các công việc này thường được thiết kế một cách tách biệt, giữa chúng thường không có mối quan hệ nào về thứ tự thực hiện, cũng như không có những ràng buộc quan trọng liên quan đến quy luật hoạt động của doanh nghiệp. Trên thực tế, những ứng dụng này thật sự giúp ích cho doanh nghiệp, nhưng xét cho cùng, vì nó không thể hiện được các quy luật kinh doanh thực tế của doanh nghiệp, nên ngày càng khó đáp ứng được các nhu cầu ngày càng cao của họ. Sự xuất hiện của các HQT LCV giúp giải quyết những vấn đề này.
- HQT LCV đồng thời hỗ trợ người dùng thực hiện công việc của mình, theo một quy luật cụ thể. Trên thực tế, các công việc được hoàn tất không độc lập với nhau mà có mối quan hệ nhất định. Công việc này phải được thực hiện trước công việc kia, để thực hiện công việc này đòi hỏi phải theo những quy định... Tất cả hoàn thành sẽ đạt được một mục tiêu nhất định. Chẳng hạn như hoàn thành việc nhận và ký hồ sơ, tổ chức thành công kì thi tuyển sinh đại học... Nó có một quy trình, một LCV cụ thể ứng với từng mục tiêu. Các LCV có thể tách biệt hoặc có trình tự thực hiện riêng... HQT LCV giúp thực thi những LCV trong doanh nghiệp, đảm bảo thứ tự và quy tắc thực hiện, đồng thời giảm thiểu những rủi ro có thể có khi thực hiện các công việc thủ công...
- HQT LCV đồng thời cung cấp chức năng quản lý LCV hỗ trợ người quản lý của doanh nghiệp nắm bắt tình hình hoạt động của các LCV nhanh chóng,

---

<sup>6</sup> Từ nay chúng tôi sẽ dùng từ viết tắt HQT LCV để chỉ Hệ Quản Trị Luồng Công Việc

kịp thời, nhằm đưa ra những xử lý thích hợp, giảm thiểu tình trạng gắp rắc rối mà không biết sớm dẫn đến công việc bị thất bại; vì thế cũng góp phần giảm thiểu rủi ro thực thi cho doanh nghiệp.

Ngoài ra, theo thời gian, các HQT LCV được các công ty lập trình thiết kế phát triển, mở rộng nhiều tính năng nên ngày càng phù hợp với quy luật hoạt động tự nhiên của doanh nghiệp, đồng thời tăng cường tối đa sự hỗ trợ, nên ngày càng trở nên quan trọng hơn.

### **3. Mô hình hóa luồng công việc và vấn đề tồn tại**

Việc MHH LCV bằng các ngôn ngữ MHH là nhằm mục đích phục vụ cho việc đưa LCV vào trong các HQT LCV. Tuy nhiên, các HQT LCV thật sự giúp ích rất nhiều cho doanh nghiệp dẫn đến một vấn nạn, đó là việc ngày càng nhiều doanh nghiệp có nhu cầu đưa LCV vào trong ứng dụng hỗ trợ dẫn tới việc phát sinh nhiều loại HQT LCV khác nhau. Các HQT này được thiết kế bởi nhiều tổ chức khác nhau nên sẽ có nhiều quy cách MHH riêng biệt đặc trưng. Điều này dẫn đến sự khó khăn cho các doanh nghiệp khi muốn thay đổi một HQT LCV này sang một HQT LCV khác, bởi họ phải thay đổi toàn bộ mô hình đã được thiết kế và thực thi, dẫn đến việc ngừng sử dụng HQT này cho đến khi thay đổi bằng 1 HQT khác phù hợp hơn.

Nhu cầu thay đổi xảy ra khi HQT LCV trở nên cũ, không còn đáp ứng nhu cầu ngày càng phát triển của doanh nghiệp, hoặc HQT LCV đó không có khả năng đáp ứng hết các yêu cầu của doanh nghiệp khiến họ phải nghĩ đến việc thay đổi một HQT LCV khác.

Đặc biệt là việc đưa vào trong HQT LCV mới các ứng dụng hỗ trợ khác nhau đáp ứng các nhu cầu khác nhau của doanh nghiệp. Đây là điều hiển nhiên bởi trên thực tế, khó có HQT nào đáp ứng hết các nhu cầu đa dạng của doanh nghiệp, nhất là trong việc xử lý các quy tắc kinh doanh hay các quy luật bản chất bên trong doanh nghiệp định ra. Nhưng các ứng dụng không tương thích hoặc không hỗ trợ

cùng một định dạng MHH dẫn đến sự bế tắc trong việc sử dụng nhiều giải pháp hỗ trợ, khiến các doanh nghiệp không thỏa mãn được hết các nhu cầu của họ.

Chính những điều kiện này đã phát sinh nhu cầu tạo ra một tiêu chuẩn quốc tế cho việc xây dựng các HQT LCV và các ngôn ngữ MHH LCV. WfMC (*Workflow Management Coalition*) ra đời nhằm mục đích này.

#### **4. Workflow Management Coalition (WfMC)**

*Workflow Management Coalition* là tổ chức thế giới (gọi tắt là WfMC) được thành lập nhằm mục đích quy định ra các tiêu chuẩn cho việc tự động hóa LCV.

Được thành lập vào tháng 8 năm 1993, hiện nay WfMC đã có hơn 200 thành viên đến từ các ngành công nghiệp và các khu nghiên cứu khác nhau. Nhiệm vụ của tổ chức WfMC là tập trung vào việc xác định các phạm vi chức năng quản lý LCV phổ biến, từ đó phát triển các chức năng này và bổ sung 1 cách thích hợp cho các ứng dụng hỗ trợ và các HQT LCV.

Cho đến nay, WfMC đã đưa ra mô hình tham chiếu chuẩn cho LCV và liên tục cải tiến, đồng thời phát triển các ngôn ngữ chuẩn cho việc MHH LCV sử dụng trong các ứng dụng tự động hóa và các HQT LCV.

##### **4.1. Mô hình tham chiếu LCV (Workflow Reference Model - WfRM)**

WfRM phát triển cấu trúc tổng quát của một ứng dụng LCV bằng cách sử dụng các giao thức (interface) cho phép sản phẩm tương tác với nhau theo nhiều cấp độ. Tất cả những HQT LCV chứa đựng nhiều thành phần khác nhau được định nghĩa theo nhiều cách đồng thời cùng một thành phần nhưng với các sản phẩm khác nhau sẽ được thể hiện khác nhau.

Hình bên dưới mô tả những thành phần và giao thức quan trọng bên trong kiến trúc LCV.

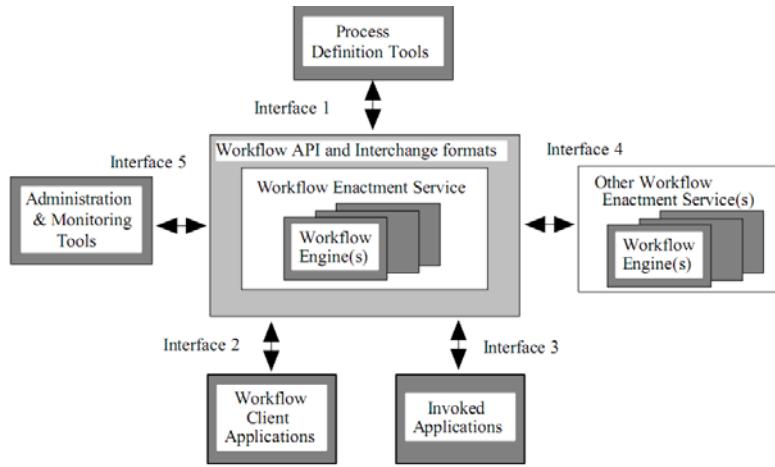


Fig 6 Workflow Reference Model - Components & Interfaces

## Hình 5. WfRM<sup>7</sup>

### - *Workflow Enactment Services (WES):*

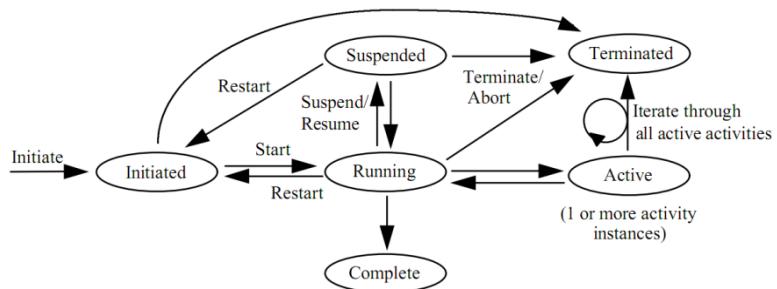
- Định nghĩa: là một dịch vụ chứa một hay nhiều *engine* để tạo ra, quản lý và thực thi các thể hiện LCV (*workflow instance*)
- Những ứng dụng bên ngoài tương tác với dịch vụ này thông qua API của LCV, gọi là WAPI<sup>8</sup>. LCV và các tác vụ được ngăn cách với nhau một cách logic. Chính sự ngăn cách này tạo nên một khả năng ứng dụng rộng rãi đối với từng loại nghiệp vụ khác nhau hoặc giữa các ứng dụng với nhau. Những tài nguyên bên ngoài được truyền vào WES bằng 1 trong 2 giao thức sau:
  - Giao thức cho các ứng dụng phía người dùng (client): trình điều khiển danh sách công việc sẽ chịu trách nhiệm cho việc chọn lựa và thực thi các công việc. Việc khởi tạo những ứng dụng cũng nằm trong sự quản lý của trình điều khiển này.

<sup>7</sup> David Hollingsworth, The Workflow Reference Model, Workflow Management Coalition Specification, 1995, trang 20

<sup>8</sup> Workflow Application Programming Interface & Interchange: thư viện chứa các API và các hàm mở rộng tùy theo chức năng của các giao thức

- Giao thức cho các ứng dụng được gọi thực thi: sẽ làm cho *Workflow Engine* có khả năng khởi động các ứng dụng chịu trách nhiệm một tác vụ cụ thể nào đó, có thể đó là một ứng dụng về phía server. Nó được gọi thực thi thông qua *Interface Worklist* nhằm đem lại sự linh động hơn cho điều phối những thẻ hiện LCV của người dùng.
- Chức năng:
  - Cung cấp môi trường thực thi để việc khởi tạo và khởi động thẻ hiện LCV xảy ra.
  - Sử dụng *Management Engine* của LCV, chịu trách nhiệm trong việc thông dịch và khởi động các thẻ hiện LCV.
  - Tương tác với những tài nguyên cần thiết để thực thi các tác vụ khác nhau.
- *Workflow Engine (WE)*:
  - Định nghĩa: là một dịch vụ, engine cung cấp môi trường để một thẻ hiện LCV thực thi. Một WES có thể chứa nhiều WE.
  - Chức năng:
    - Thông dịch sự khởi tạo của 1 thẻ hiện LCV.
    - Quản lý các thẻ hiện LCV bao gồm: tạo ra, khởi động, tạm ngưng, kết thúc v.v...
    - Đăng ký và kết thúc một LCV tham gia nhất định.
    - Nhận diện những công việc cụ thể để người dùng hoặc các giao thức để người dùng có thể can thiệp được.

- Bảo trì dữ liệu của các *Workflow Control*, các LCV liên quan và truyền tải dữ liệu đến hoặc đi từ những ứng dụng của người dùng.
  - Gọi thực thi các ứng dụng bên ngoài và kết nối những dữ liệu liên quan.
- *Homogeneous & Heterogeneous Workflow Enactment Services:*
- Homogeneous WES: bao gồm một hay nhiều WE cung cấp môi trường thực thi các LCV với các thuộc tính được định nghĩa sẵn.
  - Heterogeneous WES: bao gồm 2 hay nhiều dịch vụ khác nhau với một chuẩn mực nhất định về khả năng tương tác ở một mức độ nhất định.
  - Chức năng:
    - Hỗ trợ các LCV trong việc định nghĩa các đối tượng và thuộc tính.
    - Hỗ trợ việc vận chuyển dữ liệu liên quan.
    - Hỗ trợ LCV hoặc các tác vụ giữa các WE khác nhau.
    - Hỗ trợ những chức năng quản trị và điều khiển.
  - LCV và các trạng thái chuyển của các tác vụ:
    - WES được xem như một cái máy chuyển đổi trạng thái, nơi mà những LCV riêng rẽ hoặc các thể hiện của các tác vụ chuyển đổi trạng thái khi có những sự kiện bên ngoài tác động hoặc điều khiển những quyết định của WE.
    - Mô hình chuyển đổi trạng thái:



**Hình 6. Mô hình chuyển đổi trạng thái<sup>9</sup>**

- Initiated – thể hiện LCV được khởi tạo bao gồm những trạng thái của các LCV và những dữ liệu liên quan. Tuy nhiên ở giai đoạn này các LCV chưa hoàn toàn đầy đủ thông tin về các điều kiện để phát hiện thực thi.
- Running – thể hiện LCV bắt đầu khởi tạo.
- Active - một hoặc nhiều các tác vụ được bắt đầu.
- Suspended – thể hiện LCV dừng hoạt động và không có bất cứ một tác nào được thực hiện cho đến khi LCV trở lại trạng thái Running.
- Completed – thể hiện LCV có được đầy đủ thông tin để hoàn tất. Thể hiện bị hủy.
- Terminated – sự thực thi của các thể hiện LCV dừng trước khi hoàn tất, tất cả các những hoạt động như những ghi nhận lỗi sẽ được thông báo và thể hiện LCV bị hủy.
- *Process Definition Tools:* Là các công cụ khác nhau được sử dụng để phân tích, MHH, mô tả và ghi nhận một LCV của một nghiệp vụ nào đó. Mục đích cuối cùng từ việc MHH LCV và thiết kế tác vụ chính là định nghĩa LCV được thông dịch lúc thực thi bởi WE bên trong WES.

---

<sup>9</sup> David Hollingsworth, The Workflow Reference Model, Workflow Management Coalition Specification, 1995, trang 23

## 4.2. Các ngôn ngữ MHH LCV

WfMC đã đưa ra tiêu chuẩn nhằm thống nhất định dạng ngôn ngữ MHH chung cho các HQT LCV, nhằm giúp các doanh nghiệp thay đổi hoặc kết hợp sử dụng các phần mềm quản lý LCV khác nhau theo cách thống nhất, dễ dàng, không phải xây dựng lại khi thay đổi hay thêm phần mềm khác vào hệ thống.

Hiện nay, có 2 chuẩn đã được WfMC đề nghị là XPDL và Wf-XML.

### 4.2.1. XML Process Definition Language (XPDL):

XPDL là 1 trong 2 định dạng chuẩn được WfMC xem xét và đề nghị. Mục đích của XPDL là trao đổi các định nghĩa và thể hiện của các tiến trình nghiệp vụ (*Business Process*<sup>10</sup>), LCV giữa các sản phẩm LCV khác nhau, chẳng hạn như giữa công cụ MHH và HQT LCV. XPDL định nghĩa 1 lược đồ xml (*XML schema*) nhằm xác định phần khai báo của LCV/ Tiến trình nghiệp vụ.

Hiện nay XPDL được xem là định dạng file tốt nhất cho việc trao đổi sơ đồ BPMN (*Business Process Modelling Notation*<sup>11</sup>), cũng như sơ đồ LCV. Nó được thiết kế đặc biệt để có thể lưu trữ tất cả các tình trạng của sơ đồ BPMN. XPDL chứa các đối tượng để lưu trữ thông tin đồ họa, như vị trí X,Y của node, cũng như các tình trạng thực thi, dùng để chạy 1 thể hiện LCV. Điều này giúp phân biệt XPDL với BPEL (*Business Process Execution Language*<sup>12</sup>), chỉ tập trung vào tình trạng thực thi của thể hiện LCV. BPEL không chứa các đối tượng diễn tả thông tin đồ họa của sơ đồ LCV.

Cho đến hiện nay, trên thế giới đã có khoảng hơn 80 sản phẩm, ứng dụng sử dụng XPDL được xây dựng trên cả nền Java, Microsoft.Net Framework và Linux<sup>13</sup>.

<sup>10</sup> Xem chi tiết ở khóa luận tốt nghiệp "Tìm hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quy trình nghiệp vụ", Khoa CNTT, trường ĐHKHTN tp HCM, năm 2009

<sup>11</sup> là dạng biểu diễn đồ họa nhằm xác định Tiến trình nghiệp vụ trong LCV

<sup>12</sup> là dạng rút gọn của WS-BPEL - Web Service Business Process Execution Language - một chuẩn ngôn ngữ thực thi tiến trình xác định các tương tác với các dịch vụ web

<sup>13</sup> Xem phụ lục: Danh sách các sản phẩm, ứng dụng sử dụng XPDL

Ví dụ: 1 file mô tả 1 workflow sử dụng XPDL 2.0 có thể download tại trang web sau: <http://wfmc.org/Download-document/XPDL-Sample-Workflow-Schema.html>

#### 4.2.2. Wf-XML:

Wf-XML là 1 định dạng file tuân theo chuẩn BPM (*Business Process Management*) được phát triển bởi WfMC.

Wf-XML được thiết kế và thực thi như 1 phần mở rộng cho giao thức ASAP (*OASIS Asynchronous Service Access Protocol*) - 1 giao thức đã được chuẩn hóa cung cấp các dịch vụ bất đồng bộ , nghĩa là cung cấp cách thức để các chương trình bắt đầu, theo dõi sự thay đổi trạng thái của các chương trình hay dịch vụ khác thực thi trong khoảng thời gian dài . ASAP cung cấp cho người dùng chức năng giám sát dịch vụ đang thực thi , đồng thời thông báo cho người dùng sự thay đổi trạng thái của nó. Wf-XML đã mở rộng chức năng này từ ASAP bằng cách cung cấp thêm 1 dịch vụ mạng cho phép gửi và nhận chương trình hoặc định nghĩa của dịch vụ được cung cấp. 1 *Engine* có tính năng này sẽ có thể cung cấp 1 dịch vụ hoạt động trong khoảng thời gian dài , có thể được lập trình bằng cách cho phép cài đặt thêm các định nghĩa LCV.

Wf-XML cung cấp 1 phương thức chuẩn hóa cho 1 engine BPM<sup>14</sup> để gọi 1 thẻ hiện LCV trong 1 *engine* khác, đồng thời đợi cho thẻ hiện LCV đó hoàn tất . Vì công cụ chỉnh sửa LCV và công cụ thực thi LCV có thể được sản xuất từ nhiều nhà phát triển khác nhau , nên cần có 1 phương thức chung để trao đổi định nghĩa LCV giữa các công cụ thiết kế và *engine* thực thi), người dùng có thể kết hợp chính xác các *Process Definition Tools*<sup>15</sup> tốt nhất với *Process Execution Engine*<sup>16</sup> tương ứng theo nhu cầu.

---

<sup>14</sup> Xem chi tiết ở [http://en.wikipedia.org/wiki/Business\\_process\\_management](http://en.wikipedia.org/wiki/Business_process_management)

<sup>15</sup> Xem lại mục 4.1

<sup>16</sup> Các engine thực thi tiến trình/LCV

Wf-XML được nghiên cứu từ khoảng năm 1997 với tên gọi là SWAP (*Simple Workflow Access Protocol*) bởi các nhà phát triển như Netscape, Oracle.... Tiếp theo đó, WfMC đã tiếp bước và liên tục phát triển, cải tiến, đầu tiên là Wf-XML 1.0 và Wf-XML 1.1. Wf-XML đã được ra đời và đưa vào sử dụng trong một số sản phẩm thương mại. Phiên bản hiện nay là Wf-XML 2.0<sup>17</sup> và đang được tiếp tục nghiên cứu, phát triển. Tuy nhiên, các sản phẩm xây dựng với Wf-XML 2.0 không tương thích ngược được với các sản phẩm sử dụng Wf-XML 1.1.

### 4.3. Các loại LCV

Chuẩn WfMC định ra các loại LCV dựa trên quy tắc hoạt động của LCV và loại nghiệp vụ kinh tế đang được đề cập , bao gồm : Production, Administrative, Collaborative, và Ad-Hoc<sup>18</sup>

#### 4.3.1. *Production*

Production Quản lý 1 số lượng lớn các tác vụ tương tự nhau , nhằm tối ưu hóa năng suất nghiệp vụ . Cách thức hoạt động của LCV loại này là tự động hóa , nghĩa là các tác vụ bên trong LCV được thực hiện một cách tự động , con người chỉ tác động lên các công việc không nằm trong tiến trình đã được định nghĩa sẵn , tức là các ngoại lệ (exceptions). Như vậy, trong loại LCV này , thời gian và độ phức tạp của các sự kiện cần sự tương tác với con người được giảm thiểu ... Việc tối ưu hóa nhằm đạt chất lượng và độ chính xác cao trong loại LCV này có thể đạt được bằng cách thi hành các tác vụ có tính lặp lại cao theo cùng 1 phương pháp 1 cách liên tục. Ứng dụng của LCV thuộc loại Production là để quản lý các tiến trình có độ phức tạp cao, đặc biệt có thể kết hợp chặt chẽ với những hệ thống đang tồn tại. Tuy nhiên, xu hướng hiện nay của việc sử dụng loại LCV này là nhúng các thành phần chức năng trong LCV vào trong các ứng dụng lớn dưới vai trò như các Rules Engine. Điều này dẫn đến việc phân chia bên trong loại LCV này thành 2 loại nhỏ : Autonomous Workflow Engines và Embed Workflow. Trong đó, sự khác nhau giữa 2 loại này ở

---

<sup>17</sup> Xem phụ lục: XML Schema của Wf-XML 2.0

<sup>18</sup> Charles Plesums, Introduction to Workflow, Computer Sciences Corporation, Financial Services Group

chỗ, Autonomous Workflow bản thân nó không cần thêm các phần mềm bổ sung , còn Embed Workflow cần phải được gắn vào 1 hệ thống nào đó, chẳng hạn như hệ thống ERP<sup>19</sup>.

#### **4.3.2. Administrative**

Dễ dàng xác định LCV . Thông thường sẽ có rất nhiều thể hiện LCV cùng thực thi đồng thời, và chúng cần sử dụng 1 lượng lớn nhân viên. Thể hiện LCV luôn dc tạo ra từ trong chương trình, và nếu chương trình đó quá phức tạp, thì họ chỉ cần sử dụng chương trình khác. Điều này có nghĩa là loại LCV này rất linh hoạt trong việc sử dụng các chương trình quản lý LCV . Như vậy, tính linh hoạt ở đây quan trọng hơn năng suất, và những hệ thống theo dạng này xử lý các trường hợp mỗi giờ với cường độ thấp hơn từ 1 đến 2 lần so với các hệ thống *Production Workflow*.

#### **4.3.3. Collaborative**

Tập trung vào các hoạt động làm việc nhóm . Các nhóm cùng hoạt động với nhau để xây dựng 1 mục tiêu chung, từ những nhóm nhỏ, hướng đề tài, đến những nhóm người khác nhau có cùng 1 mục tiêu chung ... Hiệu quả của việc sử dụng mô hình LCV này để là hỗ trợ làm việc nhóm , hiện nay được xem như 1 yếu tố quan trọng trong sự thành công của các doanh nghiệp . Lợi ích của việc sử dụng Internet và www hỗ trợ liên lạc nhóm giữa các doanh nghiệp cũng là 1 thành công thực tế trong hầu hết các tổ chức . Định nghĩa LCV ở đây không cứng nhắc mà có thể thường xuyên được thay đổi , Thỉnh thoảng người ta gọi các chương trình theo loại này là *Groupware*<sup>20</sup>. Dĩ nhiên là có rất nhiều loại Groupware không được xem như 1 ứng dụng LCV loại Collaborative , chẳng hạn như *Bulletin Boards*<sup>21</sup> hay *videoconference*<sup>22</sup>.

---

<sup>19</sup> Hệ thống hoạch định tài nguyên doanh nghiệp - một hệ thống được sử dụng trong các tổ chức nhằm mục đích hoạch định tài nguyên trong tổ chức đó. Đó có thể là một doanh nghiệp, hay bất kì loại tổ chức nào khác...

<sup>20</sup> Chương trình hỗ trợ làm việc nhóm

<sup>21</sup> Bảng thông báo, thông báo các thông tin mới đến các thành viên trong nhóm

<sup>22</sup> Chương trình hỗ trợ họp nhóm bằng Video, hỗ trợ làm việc từ xa với nhau, khi các nhân viên không đang cùng ở chung một nơi, chẳng hạn như khi đang ở các quốc gia khác nhau.

#### **4.3.4. Ad-Hoc**

Cho phép người dùng tạo ra và sửa đổi định nghĩa LCV nhanh chóng và dễ dàng để đáp ứng các trường hợp phát sinh . Như thế, Ad-Hoc có thể có rất nhiều định nghĩa LCV. Ad-hoc Workflow tối đa hóa tính linh hoạt trong các lĩnh vực mà bảo mật không phải là vấn đề chính yếu . Nghĩa là , chặng hạn với các LCV loại Production thì Tổ chức , doanh nghiệp là người sở hữu LCV , còn ở Ad-Hoc thì các người sử dụng ứng dụng có thể có tiến trình riêng của họ.

### **5. Yêu cầu của một ứng dụng quản lý LCV tổng quát**

Để đáp ứng nhu cầu của doanh nghiệp, các ứng dụng hỗ trợ quản lý LCV tổng quát cần đáp ứng rất nhiều yêu cầu về nhiều mặt:

- Về hiệu suất làm việc, ứng dụng cần phải đạt hiệu suất làm việc cao, nghĩa là cùng một lúc có thể chạy song song nhiều thể hiện LCV mà không gặp trở ngại
- Về mức độ linh hoạt, ứng dụng cần phải thể hiện sự linh hoạt tốt nhất có thể. Nghĩa là ứng dụng được thiết kế sao cho có thể thích hợp với nhiều luồng LCV khác nhau. Bởi trong doanh nghiệp hiện nay, có rất nhiều LCV cùng tồn tại song song. Chẳng hạn như, trong một trường đại học có thể sẽ có những LCV như: LCV quản lý hồ sơ công văn, LCV tổ chức thi học kì, LCV tuyển sinh đại học,... Vì thế, nếu ứng với mỗi LCV khác nhau phải sử dụng một ứng dụng hỗ trợ khác nhau thì thật sự rất khó khăn cho doanh nghiệp, ngay cả trong việc quản lý các ứng dụng, chứ chưa nói gì đến việc sử dụng chúng (vì mỗi ứng dụng khác nhau sẽ có thể có cách sử dụng khác nhau, và khó thống nhất cơ sở dữ liệu chung...)
- Chia sẻ, dùng chung dữ liệu với cơ sở dữ liệu vốn có của doanh nghiệp, và tự động cập nhật thông tin tương ứng của từng thể hiện LCV

- Về khả năng hỗ trợ quản lý, ứng dụng cần hỗ trợ người quản lý sao cho họ có thể thông qua ứng dụng nắm bắt được tình hình của từng thể hiện LCV, đồng thời kiểm soát, điều phối chúng. Nghĩa là ứng dụng phải có khả năng:
  - Cho phép người quản lý có thể thấy được thông tin của từng thể hiện LCV đang thực thi ở dạng hình ảnh dễ hiểu (biểu đồ, sơ đồ...), đồng thời cho phép kiểm soát chúng (runback<sup>23</sup> 1 tác vụ đang thực thi, thấy được tình trạng cụ thể, có cảnh báo khi có vấn đề xảy ra, xem được các tài liệu đi kèm ứng với từng tác vụ con
  - Thông kê lại các kết quả thực hiện theo yêu cầu của quản lý và cho phép tự thiết kế các kiểu thống kê phù hợp...
  - Cho phép phân công, cập nhật phân công từng tác vụ sao cho hợp với từng thời điểm thực tại.
  - Gửi thông báo đến quản lý trong bất cứ tình huống xảy ra nào.
  - Cập nhật thời gian thực hiện cho phù hợp từng thời kì, từng thể hiện khác nhau theo thực tế.

Đây chỉ là những yêu cầu chung của một ứng dụng quản lý LCV tổng quát. Trên thực tế, ứng với mỗi doanh nghiệp khác nhau với nền văn hóa doanh nghiệp cũng khác nhau, sẽ có các yêu cầu riêng rất biệt mà ứng dụng thực tế cần phải thỏa mãn cả những điều này.

*Tóm lại, trong chương này, chúng tôi đã trình bày các kiến thức cơ bản về LCV cũng như MHH LCV, từ đó làm cơ sở cho những chương tiếp theo trình bày về các kỹ thuật, công nghệ mà chúng tôi đã nghiên cứu để xây dựng ứng dụng, cũng như nội dung của ứng dụng mà chúng tôi đã thực hiện trong khóa luận.*

---

<sup>23</sup> Thực hiện lại, nghĩa là khi 1 tác vụ nào đó thực hiện không đúng theo yêu cầu thực tế hoặc không đúng với thực tế, người quản lý có thể cho phép thực hiện lại tác vụ đó và gửi thông báo đến nhân viên tương ứng

## **Chương 3: Môi trường công nghệ để xây dựng một ứng dụng quản lý luồng công việc**

*Chương này đề nghị một quy cách mô hình hóa dựa trên công cụ mô hình hóa cung cấp bởi Microsoft trong bộ .Net 3.0. Quy cách này có thể thích ứng tốt với các ứng dụng quản lý cho phép người dùng tương tác trực tiếp đến các thành phần LCV trong quy trình. Tại đó có thể đưa ra các điều khiển đối với sự thực thi công việc thành phần mà người dùng có quyền truy cập. Tuy nhiên, trước khi bàn đến vấn đề này, chúng tôi xin điểm lại một số khái niệm và kiến trúc quan trọng liên quan đến việc MHH LCV. Đó cũng là nội dung chính mà chúng tôi đã nghiên cứu trong suốt thời gian thực hiện khóa luận.*

*Vì đề tài LCV đã được đề cập đến ở một các khóa luận tốt nghiệp trước<sup>24</sup>, nên chúng tôi chỉ đề cập đến những vấn đề thật sự quan trọng và cần thiết trong khóa luận này, phần còn lại sẽ bị bỏ qua hoặc có tham chiếu trực tiếp đến các khóa luận đó. Cũng cần nói thêm rằng, một trong 2 khóa luận trước nghiên cứu về mô hình LCV trên môi trường mã nguồn mở jBPM, khóa luận còn lại nghiên cứu về Windows Workflow Foundation với sự hỗ trợ của SharePoint Workflow. SharePoint Workflow là công cụ hỗ trợ tương tác giữa người dùng với các LCV, đáp ứng hầu hết các nhu cầu thiết kế cũng như xử lý của hệ thống cần xây dựng. Trong khóa luận này, chúng tôi chỉ sử dụng Windows Workflow Foundation làm môi trường thiết kế và triển khai các mô hình LCV cùng với công nghệ Windows Presentation Foundation làm nền tảng thiết kế ứng dụng. Ngoài ra, chúng tôi không sử dụng thêm công cụ hỗ trợ nào khác.*

### **1. Workflow Management System (viết tắt là WfMS):**

---

<sup>24</sup> (Khóa luận tốt nghiệp "Nghiên cứu giải pháp nguồn mở cho Workflow quản lý hồ sơ công văn - do sinh viên Võ Hữu Phúc - Đào Anh Vũ thực hiện, tháng 3 năm 2009; và đề tài "Tim hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quy trình nghiệp vụ - do sinh viên Lê Nhựt Minh - Nguyễn Trần Minh Tú thực hiện, tháng 3 năm 2009");

Tuy đã được đề cập đến trong chương 2, nhưng vì chúng tôi chưa đưa ra định nghĩa nào cho WfMS nên trong chương này chúng tôi cũng xin mạn phép đưa ra định nghĩa cho WfMS<sup>25</sup>.

WfMS (HQT LCV) là một hệ thống phần mềm hỗ trợ việc thiết kế, thực thi các LCV. Mỗi WfMS đưa ra các phương pháp định nghĩa mô hình LCV và điều khiển sự thực thi các LCV cũng như các công việc thành phần bên trong nó. Ngoài ra, WfMS còn có chức năng luân chuyển dữ liệu, phân phối các công việc tới các tác nhân dựa vào các công cụ và dịch vụ hỗ trợ.

## 2. *Windows Workflow Foundation* (viết tắt là WF)<sup>26</sup>

### 2.1. Đặc điểm

Hiện nay, trên thế giới, có rất nhiều nền tảng công nghệ được thiết kế nhằm hỗ trợ việc xây dựng các sản phẩm MHH LCV<sup>27</sup>, cũng như các phần mềm mã nguồn mở khác với mục đích tương tự. Trong đó, Windows Workflow Foundation là một nền tảng công nghệ do Microsoft nghiên cứu, xây dựng và đóng gói trong bộ .Net Framework (từ phiên bản 3.0 trở lên). WF cho phép người dùng định nghĩa, thực thi và quản LCV với các những điểm khác biệt so với các hệ thống khác như sau:

- WF cho phép điều khiển các công việc thực thi trong thời gian dài (nhiều ngày, nhiều tháng, ...).
- WF có thể chỉnh sửa động khi đang thực thi.
- WF là một phương pháp lập trình khai báo kết nối các thành phần được định nghĩa sẵn (*activity*).
- WF cho phép người dùng định nghĩa các LCV.

<sup>25</sup> Workflow Management Systems: A Survey - SHI Meilin, YANG Guangxin, XIANG Yong, WU Shangg. Department of Computer Science, Tsinghua University, Beijing, P.R.China 100084

<sup>26</sup> Microsoft Windows Workflow Foundation Step by Step - Redmond, Washington 98052-6399 Copyright © 2007 by Kenn Scribner

<sup>27</sup> Xem khóa luận "Tìm hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quy trình nghiệp vụ" - Chương 2

- WF hỗ trợ nhiều kiểu ứng dụng khác nhau.

Với WF, việc phát triển ứng dụng .Net trở nên dễ dàng hơn, đồng thời chất lượng ứng dụng cũng được nâng cao. Đây không phải là một ứng dụng độc lập, mà là một nền tảng phần mềm được thiết kế nhằm cho phép đưa LCV vào trong ứng dụng. Sức mạnh của việc sử dụng LCV kết hợp vào trong ứng dụng là ở chỗ:

- Đối với những ứng dụng hỗ trợ nghiệp vụ kinh doanh, WF giúp ta có thể kết hợp các quy trình nghiệp vụ đó.
- Đối với những ứng dụng đòi hỏi nhiều tương tác với người dùng, WF cung cấp một loại ứng dụng chuyên biệt (WF State Machine) giúp thực thi các quy tắc luận lý nhằm xử lý các tương tác đó.
- Đối với những ứng dụng có khả năng tùy chỉnh cao, WF giúp ta tách biệt giữa các thành phần nghiệp vụ kinh doanh với LCV. Điều này cho phép chúng ta có thể thay đổi được cả luồng điều khiển mà không ảnh hưởng đến những login bên dưới trong doanh nghiệp.

Đồng thời, WF cũng có một số đặc điểm ưu việt sau:

- Cung cấp một kiến trúc mạnh mẽ và linh hoạt cho việc phát triển ứng dụng LCV
- Thúc đẩy sự nhất quán trong phát triển ứng dụng, giúp cải thiện năng suất trong việc tạo ra ứng dụng mới cũng như sửa chữa, nâng cấp ứng dụng cũ
- Hỗ trợ hai kiểu định nghĩa LCV khác nhau: LCV tuần tự hay LCV trạng thái - cho phép xây dựng LCV theo hướng tương tác ứng dụng hay tương tác với người dùng .
- Cung cấp dịch vụ *Persistence*, cho phép thực thi LCV trong một thời gian dài.

- Có khả năng mở rộng vô hạn; nghĩa là Microsoft cung cấp một số điểm mở rộng cho phép chúng ta có thể sửa đổi những thuộc tính mặc định của LCV. Chẳng hạn như khi dịch vụ SQL Persistence nêu trên không đáp ứng được nhu cầu của chúng ta, chúng ta có thể tự xây dựng dịch vụ riêng của mình
- Được tích hợp vào trong .Net Framework và là thành phần miễn phí.

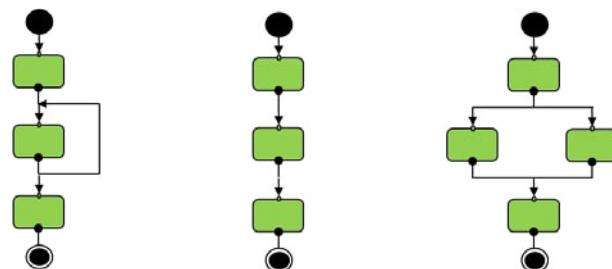
Trên đây cũng chính là những nguyên nhân đã thuyết phục chúng tôi quyết định sử dụng WF để xây dựng ứng dụng cho khóa luận này.

## 2.2. Các loại LCV trong WF:

Trong WF, người ta không chia rõ ràng các loại LCV theo như chuẩn WfMC, mà thay vào đó, WF phân loại theo loại ứng dụng LCV, chủ yếu gồm 3 loại sau:

### 2.2.1. Sequential Workflow (Workflow tuần tự):

Trong ứng dụng WF theo dạng này , các tác vụ có thể được thực thi 1 cách tự động với không có hoặc có rất ít tác động từ bên ngoài . Bản thân LCV có thể tự điều khiển việc thực thi các tác vụ. Như thế, trong loại ứng dụng này sẽ có rất ít các tương tác qua lại với người dùng . Các Activity<sup>28</sup> trong LCV sẽ được thực thi theo thứ tự người lập trình mô tả khi xây dựng ứng dụng . Đặc điểm: Các công việc sẽ được biểu diễn liên tiếp nhau. Một công việc chỉ được thực hiện khi các công việc phía trước đó đã hoàn thành



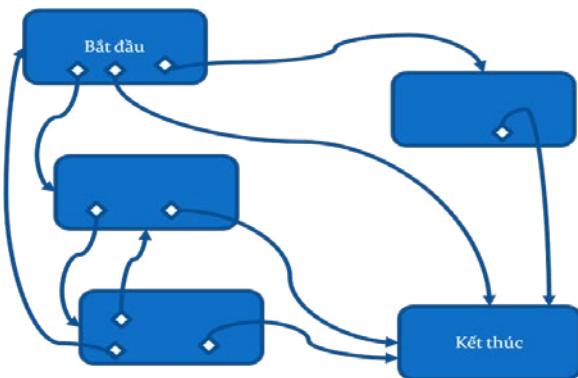
**Hình 7. Các mô hình LCV tuần tự**

---

<sup>28</sup> Đơn vị nhỏ nhất trong luồng công việc, được quy định trong WF. Chi tiết về kiến trúc LCV trong WF chúng tôi sẽ đề cập ngay trong phần sau

### 2.2.2. State-Machine Workflow (*Workflow trạng thái*):

Trong ứng dụng WF theo dạng này, LCV về cơ bản sẽ phụ thuộc vào các sự kiện bên ngoài tác động vào để có thể thực thi tác vụ kế tiếp . Như vậy, các tác vụ phụ thuộc chặt chẽ vào các tác động bên ngoài điều khiển quá trình thực thi của nó. Loại ứng dụng LCV này cần rất nhiều sự tương tác với người dùng : người dùng sẽ tác động, xác nhận hay thực thi các hành động khác nhau trong suốt LCV và ghi nhận kết quả để LCV xác định trạng thái của nó và quyết định hoạt động tiếp theo cho nó. Đặc điểm: LCV sẽ được mô hình hóa thành một tập các trạng thái cụ thể. Tại mỗi trạng thái có thể có một tập các sự kiện được phép xảy ra. Mỗi một sự kiện xảy ra sẽ dẫn tới việc chuyển đổi hệ thống từ trạng thái này sang trạng thái khác.



Hình 8. MH LCV trạng thái

### 2.2.3. Rules-based Workflow (*Workflow dựa vào quy luật*)

Các quy luật kinh doanh tồn tại giúp giải quyết các quyết định phức tạp là vấn đề thường có trong các LCV lớn, có độ phức tạp cao , nhưng bản thân Sequentia 1 Workflow hay State -machine Workflow đều không cung cấp trực tiếp cách xử lý cho các quy luật kinh doanh này . Rules-based Workflow được xây dựng dựa trên Sequential Workflow, nhưng nó là rules -driven, tức là có thêm các quy luật nhằm điều khiển các hoạt động của các tác vụ. Thay vì chỉ thực thi các hoạt động lập trình viên mô tả, loại LCV này sẽ kết hợp thêm các *Policy activity* (1 loại Activity trong

WF dùng để xây dựng quy tắc hoạt động của các Activity) và các Rule Conditions để thi hành các tác vụ dựa trên quy luật kinh doanh được mô tả trong LCV.

Tóm lại, một LCV có một trạng thái bắt đầu và một trạng thái kết thúc nhất định. Trạng thái bắt đầu là điểm khởi đầu mặc định của một quá trình thực thi quy trình. Còn trạng thái kết thúc thì tùy vào loại mô hình LCV mà có những điều kiện khác nhau để đạt được đến các trạng thái này. Ví dụ, với LCV Tuần tự thì trạng thái kết thúc sẽ được đạt đến khi tất cả các công việc thành phần đều được thực hiện xong; còn với LCV trạng thái thì có thể kết thúc bất cứ khi nào xảy ra sự kiện dẫn đến trạng thái này.

### 2.3. Kiến trúc cơ bản của WF

Kiến trúc cơ bản của WF gồm có 3 thành phần chính: BAL (*Base Activity Library*), *Runtime Engine* và *Runtime Services*; có thể được mô tả như hình sau:



Hình 9. Cấu trúc luồng công việc<sup>29</sup>

#### 2.3.1. Activity và Custom Activity

Một LCV trong WF được tạo thành từ các Activity, là các đối tượng được xây dựng sẵn tương tự các đối tượng quen thuộc (button, textbox...) trong ứng dụng dành cho Windows. Mỗi đơn vị công việc trong thực tế khi mô hình hóa sẽ thành

---

<sup>29</sup> Introduction to Workflow Foundation

một đối tượng Activity. Đây cũng chính là thành phần nhỏ nhất trong mô hình LCV định nghĩa bởi WF.

Để xây dựng ứng dụng WF, Microsoft cung cấp một thư viện các Activity cơ bản<sup>30</sup>, gọi là Base Activity Library (gọi tắt là BAL). Ngoài ra, người sử dụng có thể tự định nghĩa các Activity khác nhằm phục vụ cho nhu cầu đặc trưng của từng quy trình nghiệp vụ - gọi là các Custom Activity. Các Activity này cũng như cả LCV có thể được thiết kế, định nghĩa thông qua Workflow Designer tích hợp sẵn trong công cụ Visual Studio hoặc bằng cách viết mã XAML<sup>31</sup> trực tiếp.

### 2.3.2. WF Runtime

Cung cấp các thành phần cơ bản để thực thi và quản lý thể hiện LCV, cung cấp chức năng trả về tình trạng thực thi của thể hiện, đồng thời lưu trú các thể hiện LCV độc lập. WF Runtime gồm 2 loại là Runtime Services và Runtime Local Services

#### 2.3.2.1. Runtime Services

Cung cấp khả năng đăng ký các dịch vụ có thể được chạy bên trong nó. Mỗi LCV được thiết kế có thể đăng ký nhiều dịch vụ khác nhau. Những dịch vụ này được Microsoft thiết kế sẵn nhằm đáp ứng nhiều mục đích khác nhau của người thiết kế cũng như của bản thân LCV và ứng dụng hỗ trợ. Những dịch vụ này bao gồm:

- Handle Thread Scheduling Services: lập lịch thực thi các thể LCV
- Persistence Services: rất hữu ích trong việc thực thi LCV trong một thời gian dài. Dịch vụ này giúp lưu trữ lại tình trạng thực thi của từng thể hiện LCV vào trong hệ quản trị cơ sở dữ liệu với dữ liệu riêng được thiết kế sẵn.
- Transaction Services
- Queuing Services

---

<sup>31</sup> eXtensible Application Mark-up Language – là một ngôn ngữ dạng khái báo mới của Microsoft

- Tracking Services: Theo vết quá trình thực thi các công việc trong LCV
- CommitWorkBatch Services

Tuy nhiên, nếu những dịch vụ này chưa đáp ứng đủ nhu cầu của LCV cũng như ứng dụng hỗ trợ, người thiết kế hoàn toàn có thể tự xây dựng dịch vụ riêng và đăng kí với WF Runtime trước khi thể hiện LCV được khởi tạo và thực thi.

Vì LCV được thiết kế trong ứng dụng cần được thực thi trong khoảng thời gian dài, chúng tôi đã quyết định nghiên cứu và sử dụng Persistence Service, đồng thời thiết kế cơ sở dữ liệu riêng để lưu vết quá trình thực thi, mà không dùng các dịch vụ khác.

### **2.3.2.2. WF Runtime Local Services**

WF Runtime cho phép những dịch vụ do người thiết kế tạo ra chạy bên trong nó, những dịch vụ này được gọi là WF Runtime Local Services. Một dịch vụ cục bộ như thế có thể được thiết kế nhằm đáp ứng bất kì mục đích nào, chẳng hạn như cho phép thể hiện LCV có thể giao tiếp với ứng dụng chủ.

#### **2.4. Thiết kế LCV trong WF**

Khác với nhiều phần mềm LCV khác, WF không hỗ trợ XPDL hay Wf -XML. Thay vào đó, LCV trong WF được thiết kế bằng các công cụ design (Workflow Design Tools) được tích hợp vào bộ Visual Studio 2008 (với Visual Studio 2005 cần phải cài đặt thêm các thành phần bổ sung), đồng thời tự động phát sinh ra file thiết kế với 2 định dạng: C# (hoặc VB) tương thích với nền .net 2.5 framework trở lên, và đặc biệt là XAML trên nền .Net3.0 trở lên.

##### **2.4.1. C#, VB.net:**

Khi thiết kế LCV với WF, mỗi LCV được mô tả bởi 2 file: file \*.cs xử lý các sự kiện bên trong LCV, và file \*.designer.cs (với VB.net tương ứng là file \*.vb và \*.designer.vb, mô tả sơ đồ các activity bên trong LCV đó. File này được tự động

phát sinh bởi trình biên dịch (Visual studio ) tương tự như file design của Form trong 1 ứng dụng Windows.

Khi lập trình viên thêm mới 1 activity vào LCV, trình biên dịch sẽ cập nhật thông tin của activity này vào trong file designer (phát sinh code bằng C # hoặc VB.net). Cũng tương tự như project WinForm vậy , ở đây, ta có thể xem 1 LCV như 1 Form. Trong đó xử lý các LCV là các event , còn các Activity trong LCV chính là các control trong Windows Form.<sup>32</sup>

#### **2.4.2. XAML:**

Ngoài cách sử dụng C # hay VB .net để định nghĩa LCV (thường được gọi là *imperative definition*), WF còn có khả năng thực thi LCV dựa trên *declarative definition*, nghĩa là định nghĩa LCV bằng ngôn ngữ Markup XML (cách này giống với Wf-XML do WfMC phát triển ). Tuy nhiên, WF không hỗ trợ Wf -XML (để có thể tận dụng cấu trúc chuẩn của W fMC Wf-XML, lập trình viên cần phải tự xây dựng bộ biên dịch để chuyển đổi cấu trúc Wf -XML thành dạng Workflow Runtime có thể hiểu và thực thi).

Mỗi cách đều có những lợi điểm và nhược điểm riêng của nó . Việc sử dụng C#, VB.Net sẽ giúp chương trình thực thi nhanh hơn. Tuy nhiên, điểm yếu của nó chính là việc LCV phải được định nghĩa lúc buildtime. Nếu có sự thay đổi trong tiến trình nghiệp vụ, chương trình cần được thiết kế lại , biên dịch lại (ngoại trừ trường hợp chương trình có sử dụng các Rules Condition cho phép thay đổi Business Process theo 1 quy luật cụ thể nào trước đó ). Điều này khiến cho LCV trở nên thiếu linh hoạt. Trong khi nhu cầu thay đổi LCV đối với các doanh nghiệp ngày càng cần thiết (nhất là những doanh nghiệp có tiến trình nghiệp vụ thay đổi theo thời gian ). Việc sử dụng XML để định nghĩa LCV có thể giải quyết được vấn đề này.

---

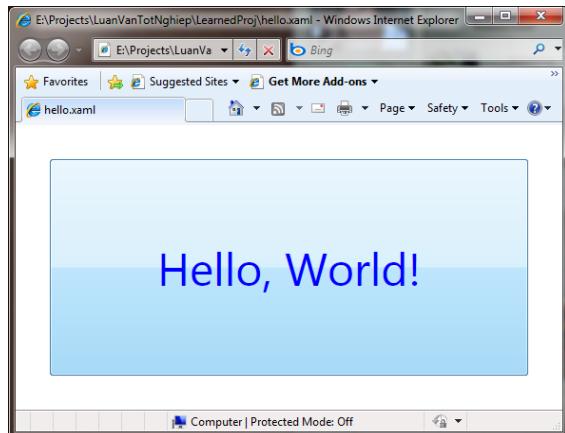
<sup>32</sup> Xem phụ lục: Định dạng mẫu tập tin Designer của WF

Vì Workflow Runtime có thể chấp nhận gần như tất cả các dạng định nghĩa LCV (phụ thuộc vào người lập trình ), tức là lập trình viên chỉ cần thông dịch định nghĩa LCV được cung cấp thành định dạng mà Workflow Runtime có thể hiểu và thực thi. Tuy nhiên, điều may mắn ở đây là WF hỗ trợ định nghĩa LCV sử dụng ngôn ngữ nền tảng xml (XML-based Workflow Definition ), đó là XAML (Extensive Application Markup Language). Việc định nghĩa LCV theo cấu trúc xml giúp LCV có thể dễ dàng được sửa đổi và triển khai . Thay vì phải biên dịch lại trong Visual Studio, người dùng chỉ cần chỉnh sửa lại file định nghĩa LCV bằng bất kỳ trình Editor nào (ngay cả notepad ), và đưa vào Workflow Runtime trước khi nó tạo lại mô hình LCV.

XAML ban đầu được đưa vào sử dụng trong WPF (Windows Presentation Foundation, 1 công nghệ được Microsoft đưa vào sử dụng từ .Net Framework 3.0 trở đi, trong đó giao diện chương trình thiết kế bằng WPF được định nghĩa thông qua file XAML). 1 file \*.xaml về bản chất không cần phải được biên dịch , mà bản thân nó có thể chạy được trên bất kỳ trình duyệt nào ở bất kỳ máy tính Windows nào có cài đặt .Net 3.0 trở lên. Chẳng hạn như , với file hello.xaml sau đây sẽ hiển thị 1 button có chữ Helloworld:

```
<?xml version="1.0"?>
<Button
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
Margin="36" Foreground="Blue" FontSize="36pt"> Hello, World!
</Button>
```

Dùng trình duyệt mở File hello.xaml sẽ hiển thị kết quả như sau:



**Hình 10. HelloWorld**

WF đã tận dụng khái niệm này từ WPF và đưa XAML vào WF . Tuy nhiên, dù WF XML được thiết kế theo XAML, định dạng file WF XML ở đây được đổi thành \*.xoml, để các công cụ tự động hóa có thể hiểu đây là 1 file mô tả LCV chứ không phải là 1 file trình bày giao diện.

## 2.5. Sự lưu trú của WF trên ứng dụng:

WF không phải là ứng dụng chạy độc lập nên nó cần được lưu trú trong một ứng dụng chủ (*host application*).NET nào đó như Winform, ASP.NET, Console, Web Service... Với đặc tính này, WF chỉ cần tập trung vào xử lý các nghiệp vụ, những phần còn lại được giao cho ứng dụng chủ xử lý.

Để tương tác với ứng dụng chủ WF cung cấp cơ chế cho phép truyền tải dữ liệu vào ra từ LCV cùng với các phương thức để điều khiển sự kiện nhằm tương tác với bên ngoài.

Dưới đây là danh sách một số dịch vụ quan trọng nhằm phục vụ cho sự lưu trú cũng như thực thi LCV trong ứng dụng chủ:

- Persistence :

- Mô tả : Là cơ chế cho phép lưu trữ LCV xuống đĩa nhớ một cách bền bỉ và khôi phục lại bất cứ khi nào. Nhờ cơ chế này, các LCV có thể được

thực thi trong một khoảng thời gian dài (nhiều ngày, nhiều tháng...) đồng thời làm giảm gánh nặng cho hệ thống phần cứng khi có nhiều LCV cùng lưu trú trong các ứng dụng.

- Các lớp đối tượng : SqlWorkflowPersistenceService
- Data Exchange: Quản lý các dịch vụ truyền thông tùy biến
- Dịch vụ tùy biến (của người dùng): WF cho phép người dùng định nghĩa các dịch vụ của riêng họ và được sử dụng như các dịch vụ chuẩn được cung cấp sẵn trong framework

Vì vậy, trong ứng dụng được xây dựng ở khóa luận này, chúng tôi đã lựa chọn giữa nhiều ứng dụng chủ khác nhau để thiết kế hệ thống. Vì sự phức tạp và nhu cầu đồ họa tương tác cao của nhiều chức năng quan trọng, chúng tôi đã quyết định sử dụng một dòng công nghệ khác, chuyên về đặc tả, thiết kế giao diện, đồng thời cũng được tích hợp trong bộ .Net 3.0 trở lên cùng với WF, đó là WPF; và chúng tôi sẽ trình bày một số kiến thức cơ bản về WPF ngay trong phần sau.

### 3. Sự khác nhau giữa WfMC và WF

#### 3.1. Về ngôn ngữ MHH LCV

Với WfMC, rõ ràng tổ chức này đã định ra 2 chuẩn định dạng lưu trữ lại LCV nhằm giúp cho các phần mềm LCV mã nguồn mở khác nhau có thể cùng trao đổi 1 thiết kế chung, đáp ứng xu hướng hiện nay của doanh nghiệp, cả về nhu cầu chất lượng LCV (LCV có xu hướng càng phức tạp hơn), về vấn đề theo dõi tình trạng, tiến độ công việc cũng như thay đổi các Business Process Definition ngay trong thời gian thực thi (Workflow runtime) ứng với những doanh nghiệp có các nghiệp vụ thay đổi theo thời gian, theo đối tượng công việc.... Cả 2 chuẩn XPDL và Wf-XML bổ sung cho nhau, có khả năng đáp ứng các yêu cầu chung của các doanh nghiệp hiện nay và trong tương lai gần. Sử dụng XPDL và Wf-XML sẽ giúp doanh nghiệp

có sự linh động hơn trong việc chọn lựa sử dụng các ứng dụng LCV, xây dựng HQT LCV, thay đổi, nâng cấp ứng dụng đang sử dụng hiện tại mà không sợ mất dữ liệu cũng như mọi thông tin về tình trạng hiện tại của hệ thống.

Đáng tiếc là cho đến nay, WF do Microsoft phát triển vẫn không hề hỗ trợ bất kỳ định dạng nào do WfMC đề ra một cách trực tiếp . Mặc dù Workflow Runtime của WF có thể chấp nhận hầu hết các dạng định nghĩa LCV, đặc biệt là với các định nghĩa được thiết kế theo định dạng xml. Vì vậy nên, về cơ bản người dùng có thể sử dụng XPDL và Wf-XML trong WF, tuy nhiên, do WF không xây dựng sẵn, nên lập trình viên cần xây dựng bộ biên dịch riêng để chuyển đổi mô hình LCV được định nghĩa bằng XPDL hay Wf-XML thành mô hình LCV trong WF.

Vì bản chất không hỗ trợ sẵn XPDL hay Wf-XML nên WF giải quyết các vấn đề doanh nghiệp theo 1 cách khác - sử dụng nhiều dịch vụ khác nhau để xử lý từng nhu cầu của doanh nghiệp. Bản chất WF là 1 nền tảng công nghệ, nghĩa là nó cung cấp 1 chuẩn chung cho các sản phẩm LCV trên Windows. Các ứng dụng, engine sử dụng công nghệ này cũng chia sẻ cùng 1 cách thức chung do Microsoft xây dựng. Cụ thể:

- XAML chỉ lưu trữ phần giao diện của WF , và danh sách các xử lý của từng activity tương ứng (cũng như trong WPF , XAML chỉ lưu trữ thiết kế giao diện và danh sách các Event ứng với các Controls chứ không lưu trữ các xử lý Event, việc xử lý được thực hiện ở file xử lý của form tương ứng).
- Vấn đề lưu trữ tình trạng hệ thống được WF xử lý bằng các sử dụng 1 dịch vụ khác, gọi là TrackingService, sử dụng SQLTrackingService để lưu lại vết quá trình thực thi vào cơ sở dữ liệu.

### **3.2. Về phân loại LCV**

WF hoàn toàn không phân loại LCV theo cách mà tổ chức WfMC phân loại. Trong đó, WfMC phân loại LCV theo đặc tính của nghiệp vụ kinh doanh tương ứng, còn WF phân loại theo mức độ tương tác với người sử dụng hệ thống. Điều

này là hợp lý, bởi WfMC là chuẩn chung, không chỉ cung cấp cho các nhà thiết kế ứng dụng, mà còn là tiêu chuẩn để các nhà kinh doanh MHH LCV tương ứng trong nghiệp vụ để phục vụ từng mục tiêu cụ thể; trong khi WF là một công nghệ hỗ trợ xây dựng ứng dụng, nên dĩ nhiên sẽ có cách phân loại riêng. Tuy nhiên, ứng với từng loại LCV cụ thể theo WfMC, hoàn toàn có thể triển khai ứng dụng tương ứng với LCV được thiết kế sử dụng WF, phụ thuộc vào LCV đó có cần nhiều sự tương tác hay không, và với mỗi tác vụ, có các quy luật phức tạp nào được định nghĩa cho nó hay không, để tương ứng xác định trong WF loại ứng dụng được dùng để khởi tạo và xây dựng LCV.

### **3.2.1. *Production Workflow:***

Với dạng LCV này (được mô tả trong chuẩn WfMC ), LCV bao gồm nhiều thao tác giống nhau lặp đi lặp lại nhiều lần, giống như quá trình sản xuất 1 sản phẩm của phân xưởng. Như vậy, với WF, chẳng hạn khi xây dựng 1 LCV mô tả qui trình sản xuất sản phẩm, lập trình viên có thể sử dụng các Activity như Parallel Activity , hay WhileActivity để biểu diễn quá trình sản xuất , phụ thuộc vào quy tắc của phân xưởng: sản xuất đồng loạt (parallel) hay tuần tự (while)...

### **3.2.2. *Administrative Workflow:***

Với dạng này , tổ chức có thể xây dựng 1 hệ thống LCV bao gồm nhiều phần mềm khác nhau cùng tuân thủ 1 chuẩn chung (Theo WfMC). Như vậy , tương tự trong WF, với mỗi mô hình LCV, người lập trình có thể chọn sử dụng các Engine nào đó được xây dựng trên cùng nền tảng WF có khả năng đáp ứng quy luật kinh doanh của tổ chức . Tuy nhiên, 1 hướng tiếp cận khác , là xây dựng LCV theo loại thứ 3 của WF , tức là dạng rule -based, khi đó , các quy luật kinh doanh phức tạp trong công ty có thể được giải quyết, thông qua việc sử dụng các Policy Activity và các Rule Conditions.

### **3.2.3. *Collaborative Workflow:***

Với dạng này, có thể sử dụng Sharepoint Workflow của Microsoft để giải quyết<sup>33</sup>

### **3.2.4. Ad-hoc:**

Với sự phát triển hiện nay của WF, các mô hình LCV có thể được chỉnh sửa trực tiếp trên giao diện, có thể thay đổi cho phù hợp với quy luật kinh doanh. Kết hợp với sự phân quyền của từng người sử dụng, có thể cho phép các người sử dụng có thể chỉ sửa đổi được các phần LCV nằm trong phạm vi hoạt động của mình, nhưng vẫn đảm bảo đầu ra cho tác vụ tiếp theo của người dùng khác.

## **4. Windows Presentation Foundation (WPF)**

### **4.1. Tổng quan**

Windows Presentaion Foundation (viết tắt là WPF) có mã là Avalon, là một bước tiến quan trọng của Microsoft trong việc xây dựng các ứng dụng của windows với giao diện lồng lẫy. WPF là thư viện lập trình giao diện đồ họa có trong Microsoft .NET Framework từ phiên bản 3.0 - thế hệ kế tiếp của WinForms. Được xây dựng trên nền Direct3D – đồ họa ba chiều, WPF phát huy tối đa sức mạnh xử lý của card đồ họa, hỗ trợ các cải tiến về giao diện trong Windows Vista, Windows 7 và độc lập với mọi độ phân giải của màn hình. WPF hướng đến sự hợp nhất của một ứng dụng dịch vụ, kế thừa và mở rộng các đặc trưng phát triển ứng dụng bao gồm: ngôn ngữ XAML (eXtensible Application Markup Language) - một loại mã đơn giản gần giống như HTML dùng để tạo và tinh chỉnh các đối tượng đồ họa, các điều khiển (control), liên kết dữ liệu (data binding), bộ cục, giao diện đồ họa 2D và 3D, các văn bản cố định và động, các mô hình đồ họa nâng cao, đồ họa vector cho phép hình ảnh tự điều chỉnh kích thước phù hợp với độ phân giải, hình ảnh động, hoạt hình, kiểu dáng (style), mẫu (template), đa phương tiện (Media), Audio và Video, văn bản và in ấn. Một ứng dụng WPF có thể được biên dịch để chạy trên máy tính hoặc 1 thiết bị thông qua trình duyệt web phổ biến như IE, Firefox, Safari... Do đó

---

<sup>33</sup> Xem chi tiết ở luận văn “Tìm hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quá trình nghiệp vụ”

WPF không chỉ dành cho các ứng dụng trên máy tính, nó còn cho phép phát triển các ứng dụng web phong phú (Rich Internet Application – RIA). Ngoài ra WPF cung cấp một mô hình lập trình nhất quán cho việc phát triển ứng dụng và cung cấp sự tách biệt rõ ràng giữa giao diện người dùng và logic nghiệp vụ.

## 4.2. XAML

XAML (*eXtensible Application Markup Language*) là một ngôn ngữ đánh dấu được sử dụng để tạo ra các đối tượng trong .NET. XAML được tạo ra để xây dựng giao diện trong các ứng dụng WPF. Ta có thể tạo ra giao diện đồ họa với khai báo thông qua XAML và dùng mã lệnh (code-behind) để tạo ra những sự kiện điều khiển những đối tượng mà ta đã định nghĩa sẵn trong XAML. Với những đòi hỏi yêu cầu cao về giao diện thì phương pháp hiệu quả nhất là tách biệt việc thiết kế giao diện và viết mã lệnh ứng dụng. Với phương pháp này thì người thiết kế và người phát triển cho ứng dụng sẽ làm việc độc lập với nhau. Với XAML thì người thiết kế và người phát triển ứng dụng hoàn toàn có thể làm việc song song với nhau. Và với *Microsoft Expression Studio*, bộ công cụ hỗ trợ trong việc thiết kế giao diện trên nền tảng XAML, cùng với *Microsoft Visual Studio* hỗ trợ cho phần *code-behind* của XAML để xử lý sự kiện cho các đối tượng mang đến sự kết hợp hoàn hảo cho người thiết kế và người phát triển ứng dụng.

## 4.3. Những tính năng nổi bật

### 4.3.1. Data Binding (liên kết dữ liệu)

Là phương pháp kết nối để lấy dữ liệu từ trong cơ sở dữ liệu để hiển thị trên giao diện. Có ba loại liên kết dữ liệu.

- One Time: dữ liệu được lấy lên từ cơ sở dữ liệu đúng một lần và không được cập nhật lại trên giao diện.
- One Way: dữ liệu chỉ được ảnh hưởng một chiều (từ một Control này đến một Control khác).

- Two Way: dữ liệu được truy cập và cập nhật từ hai chiều (từ một Control này đến một Control khác).

#### **4.3.2. Layout**

- StackPanel: Bố trí các phần tử con nằm theo chiều dọc và chiều ngang theo thứ tự mà các chúng được khai báo trong XAML.
- WrapPanel: Bố trí các phần tử con từ trái sang phải và từ trên xuống dưới.
- DockPanel: Bố trí các phần tử liên tiếp nhau và bám vào các lề của DockPanel (lề trên, lề dưới, lề trái, lề phải)
- Canvas: cho phép sắp xếp các phần tử con theo vị trí bằng cách đặt các thuộc tính Top, Left, Bottom, Right. Các phần tử con trong Canvas có thể được xếp chồng lên nhau tùy theo khai báo của chúng trong XAML.
- Grid: cho phép chúng ta phân chia dòng và cột theo một mạng lưới ô và sắp đặt các phần tử con vào các ô tùy ý.

#### **4.3.3. Resource**

Trong WPF có hai loại Resources

- Assembly Resource: Loại tài nguyên này được sử dụng trong trường hợp lưu tài nguyên hình ảnh và âm thanh để có thể sẵn sàng truy cập.
- Object Resource: Là loại tài nguyên chứa những đối tượng được định nghĩa tại một vị trí nào đó để được sử dụng trong toàn bộ chương trình.

#### **4.3.4. Style**

Là các loại thuộc tính được sử dụng trong một đối tượng đồ họa tương tự như CSS trong HTML. Style cho phép định nghĩa các tập hợp định dạng chung dùng cho toàn bộ chương trình đảm bảo tính nhất quán.

#### **4.3.5. Transformation**

Transformation là sự biến đổi hệ tọa độ cho phép đưa hình ảnh từ tọa độ này sang tọa độ khác để làm hình ảnh khác đi. Phép chuyển đổi tọa độ này được thực hiện thông qua ma trận. Mỗi phép biến đổi tọa độ như phóng to, thu nhỏ, tạo ảnh đối xứng, dịch chuyển và xoay đều có một ma trận đặc trưng cho nó.

#### **4.3.6. Animation**

Có ba loại cơ bản

- Linear interpolation: là tác động từ từ lên một thuộc tính của một phần tử nào đó tính từ điểm bắt đầu đến điểm kết thúc.
- Key Frame: Được sử dụng trên đối tượng tham chiếu hay các dữ liệu chuỗi.
- Path-based: Là hiệu ứng chuyển động theo quỹ đạo cho sẵn.

### **4.4. Microsoft Silverlight (tên mã WPF/E)**

Microsoft Silverlight là công nghệ hoạt động được trên tất cả các nền tảng, tất cả trình duyệt dựa trên nền tảng công nghệ WPF. Silverlight cung cấp một môi trường đồ họa mới hơn so với những trang web sử dụng HTML và JavaScript truyền thống. Silverlight hoạt động tương tự Flash và các ứng dụng di động với các chương trình ứng dụng như một ứng dụng .NET. Silverlight đã hỗ trợ tương tác đồ họa với những hiệu ứng vector, hiển thị hình ảnh, âm thanh và video.

*Tóm lại, WF là một nền tảng do Microsoft cung cấp với rất nhiều dịch vụ khác nhau, giúp người thiết kế có thể xây dựng được phần mềm đáp ứng các yêu cầu đa dạng của khách hàng. Đến đây, chúng ta đã bàn khá đầy đủ về các khái niệm và các kiến trúc quan trọng liên quan đến việc MHH LCV. Chương tiếp theo sẽ đi vào hiện thực hóa ứng dụng quản lý LCV, sử dụng kiến trúc WF và công nghệ WPF vừa được trình bày ở chương này.*

## **Chương 4: Hiện thực ứng dụng**

*Tiếp theo, chương này sẽ giới thiệu tổng thể về hiện thực ứng dụng mà nhóm chúng tôi đã thực hiện. Chương này sẽ gồm hai phần chính.*

*Trong phần đầu của chương, chúng tôi sẽ trình bày kết quả khảo sát bối cảnh quản lý công tác tổ chức thi, cấp chứng chỉ tin học Quốc gia và các yêu cầu cụ thể đối với ứng dụng LCV phục vụ cho lĩnh vực này.*

*Sau đó, trong phần hai của chương, chúng tôi sẽ thể hiện lại kết quả phân tích và thiết kế một ứng dụng cụ thể, dựa trên những tiêu chuẩn đã mô tả ở chương 2, bao gồm kết quả phân tích ứng dụng ở góc độ dữ liệu, kết quả thiết kế xử lý, giao diện và các giải pháp được đề xuất nhằm giải quyết các vấn đề gặp phải trong quá trình xây dựng ứng dụng.*

### **1. Tổng quan về LCV tổ chức thi, cấp chứng chỉ quốc gia**

Như đã đề cập, việc quản lý công tác tổ chức thi, tuyển sinh ở Việt Nam là thật sự quan trọng và hầu như không cho phép xảy ra bất cứ sai sót nào. Trong khóa luận này, chúng tôi chỉ đề cập tới một quy trình điển hình nhất đó là quy trình tổ chức thi, cấp chứng chỉ tin học Quốc gia<sup>34</sup>.

Một quy trình thi chuẩn bao gồm nhiều công việc xảy ra đồng thời hoặc nối tiếp nhau, được đặc trưng bởi ngày thi và địa điểm tổ chức thi. Địa điểm thi có thể là ngay tại đơn vị tổ chức thi và cấp chứng chỉ, cũng có thể là tại các đơn vị khác có liên kết đào tạo với đơn vị này<sup>35</sup>. Một công việc có thể do một hoặc nhiều nhân viên thửa hành (là nhân viên trực tiếp thực hiện công việc)<sup>36</sup> cùng phụ trách, đồng

---

<sup>34</sup> Từ đây về sau, chúng tôi dùng cụm từ “quy trình thi” để chỉ quy trình tổ chức thi và cấp chứng chỉ tin học Quốc gia

<sup>35</sup> Trong khóa luận này, chúng tôi dùng cụm từ “Cơ sở liên kết” (Viết tắt là CSLK) để chỉ đến những đơn vị liên kết đào tạo mà đợt thi diễn ra.

<sup>36</sup> Sau đây gọi tắt là Nhân viên.

thời mỗi nhân viên có thể được giao phụ trách một hoặc nhiều công việc khác nhau. Ngoài ra, quy trình thi có thể chịu sự chi phối của các tổ chức khác (tác nhân ngoài) tại các công đoạn thực hiện các công việc đặc thù.

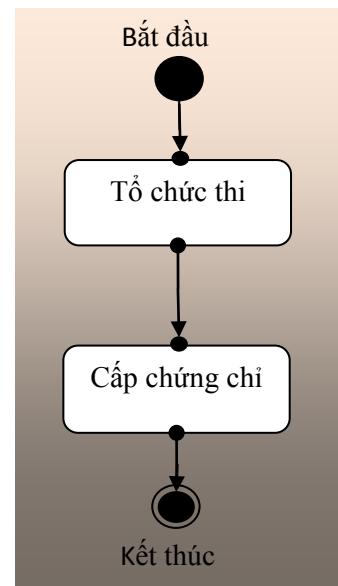
### 1.1. Các quy trình ứng dụng

Một quy trình thi bao gồm 2 giai đoạn chính: *Tổ chức thi* và *Cấp chứng chỉ*. Trong thực tế, hai giai đoạn này gắn kết với nhau thành một quá trình xuyên suốt như hình vẽ H1. Tuy nhiên, để dễ dàng trong việc mô hình hóa và mô tả chi tiết các nghiệp vụ, mỗi giai đoạn đó sẽ được “mịn hóa” thành một quy trình con (Hình H2. và H3.).

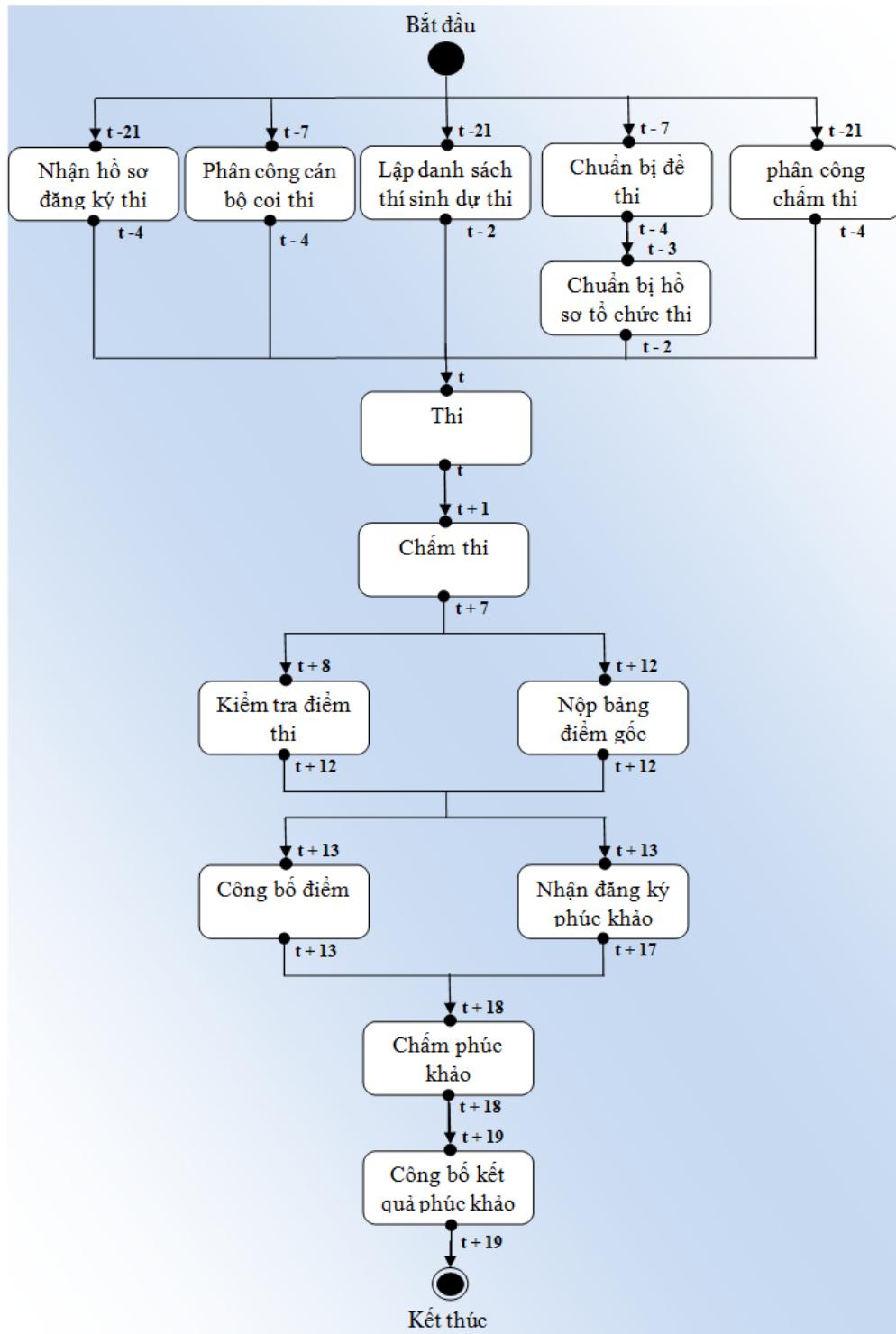
Giai đoạn Tổ chức thi kéo dài từ thời điểm bắt đầu nhận hồ sơ đăng ký dự thi của thí sinh đến thời điểm công bố điểm thi phúc khảo. Khi giai đoạn này kết thúc, danh sách thí sinh thi đạt sẽ được duyệt và xin cấp chứng chỉ. Khi đó, giai đoạn Cấp chứng chỉ được bắt đầu và kéo dài cho đến khi các chứng chỉ đã được hoàn tất và sẵn sàng cấp.

#### 1.1.1. Quy trình tổ chức thi

Hình vẽ H2. mô tả một cách chi tiết các công việc cần được thực hiện để phục vụ cho quy trình tổ chức thi. Mỗi ô hình chữ nhật trong sơ đồ thể hiện cho một đơn vị công việc, kèm theo đó là thông tin về ngày bắt đầu và ngày kết thúc đơn vị công việc đó.



**Hình 11. Quy Trình Thi và cấp chứng chỉ**



**Hình 12. Quy trình tổ chức thi**

Như đã trình bày ở trên, ngày thi là thông tin đặc trưng quan trọng nhất cho mỗi đợt thi. Và theo đó, trong mô hình luồng công việc, đơn vị công việc Thi cũng là

công việc quan trọng nhất. Gọi  $t$  là ngày thi, khi đó lấy  $t$  làm chuẩn sẽ tính được tất cả các mốc thời gian cho các đơn vị công việc khác. Ví dụ, công việc Nhận hồ sơ đăng ký thi có mốc thời gian bắt đầu là  $t - 21$  mang ý nghĩa: thời điểm bắt đầu nhận hồ sơ đăng ký thi là ngày thứ 21 trước ngày thi; Hoặc công việc Chấm thi có mốc thời gian kết thúc là  $t + 7$  mang ý nghĩa: công việc chấm thi sẽ kết thúc muộn nhất là ngày thứ 7 sau ngày thi v.v...

Trong hình vẽ H2, những công việc diễn ra song song một phần hoặc song song toàn bộ sẽ được biểu diễn ngang hàng nhau. Cụ thể hơn, công việc Nhận hồ sơ đăng ký thi và Phân công chấm thi là song song hoàn toàn, chúng được thực hiện trong cùng một khoảng thời gian từ ngày  $t - 21$  đến  $t - 4$ ; trong khi đó công việc Kiểm tra điểm thi và Nộp bảng điểm gốc là song song một phần (chỉ diễn ra đồng thời trong ngày  $t + 12$ ). Ngoài ra, với các công việc diễn ra nối tiếp nhau sẽ được mô hình hóa dạng nối tiếp, ví dụ: công việc Thi, Chấm thi...

Đến đây, cũng cần nhấn mạnh lại rằng một mô hình LCV được mô hình hóa từ các yêu cầu thực tiễn và sự thực hiện các quy trình nghiệp vụ tương ứng. Việc sắp xếp thực hiện một số công việc song song nhau nhằm mục đích rút ngắn tối đa tổng thời gian thực hiện của cả quy trình thi. Tuy nhiên cũng có một số công việc (mà theo tính chất của công việc đó) bắt buộc phải thực hiện tuần tự nhau. Trong những trường hợp này, thông thường sự kết thúc công việc trước là điều kiện cần và đủ để bắt đầu công việc sau. Cụ thể hơn, trong LCV của quy trình tổ chức thi, công việc Chấm thi không thể diễn ra trước hoặc đồng thời với công việc Thi được, vì như vậy là phi thực tế.

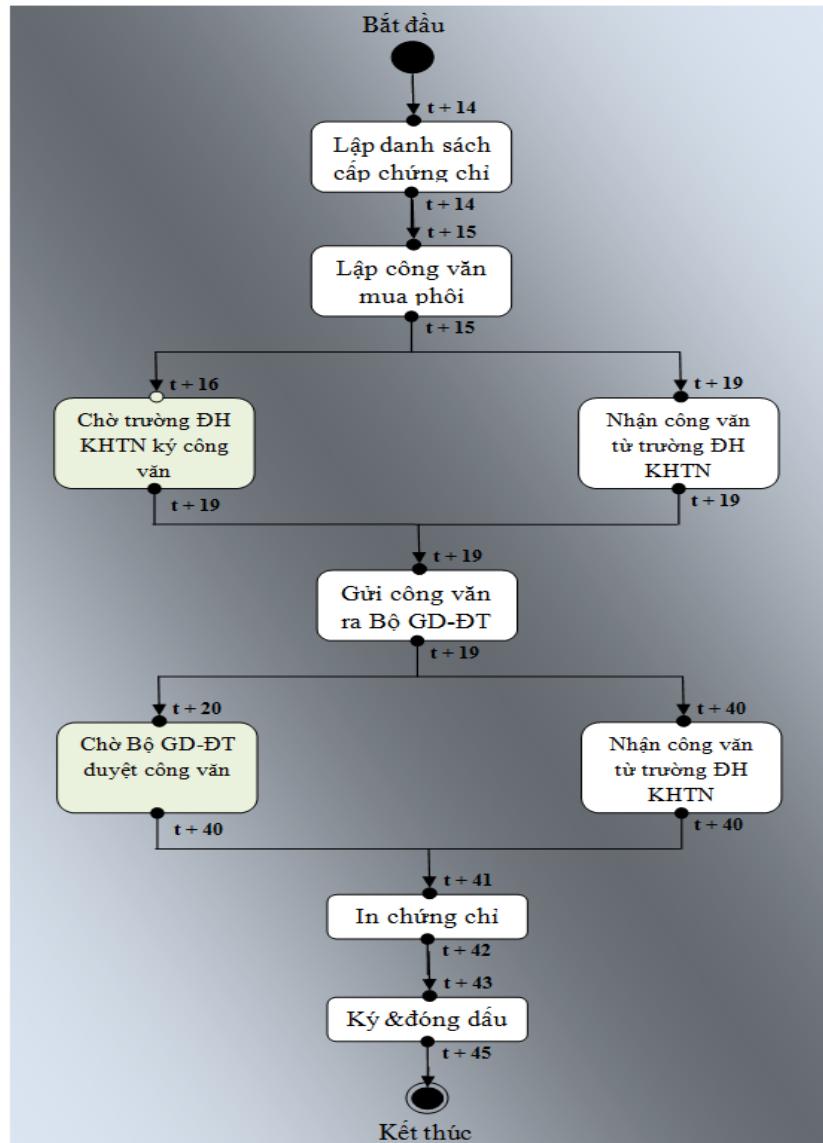
### 1.1.2. Quy trình cấp chứng chỉ

Quy trình *Cấp chứng chỉ* không có nhiều công việc phức tạp như quy trình *Tổ chức thi*, tuy nhiên lại phụ thuộc rất nhiều vào các ứng dụng bên ngoài. Chính vì lý do này, các mốc thời gian trong quy trình này hầu hết là tương đối. Việc thực thi các

công việc chịu sự chi phí của ứng dụng bên ngoài có ảnh hưởng rất lớn đến tiến độ của quy trình.

Sau khi công bố điểm thi, chậm nhất là ngày  $t + 14$ , danh sách các thí sinh thi đạt sẽ được lập. Cùng với đó, công văn mua phôi chứng chỉ cũng sẽ được lập để gửi lên giám đốc hoặc hiệu trưởng đơn vị tổ chức thi ký rồi gửi lên Bộ Giáo Dục & Đào Tạo (GD-ĐT). Sau đó tiếp tục gửi công văn đến bộ GD-ĐT và chờ duyệt mua phôi chứng chỉ. Đơn vị tổ chức thi không làm chủ được về thời gian để Hiệu trưởng và Bộ GD-ĐT ký và duyệt bán phôi chứng chỉ.

Thông thường, vào ngày  $t + 40$  là phôi chứng chỉ được chuyển về tới đơn vị tổ chức thi, sẵn sàng để in, ký tên và đóng dấu. Quá trình này diễn ra tới ngày  $t + 45$ . Quy trình *Cấp chứng chỉ* kết thúc khi các chứng chỉ đã sẵn sàng để được cấp cho thí sinh thi đạt.



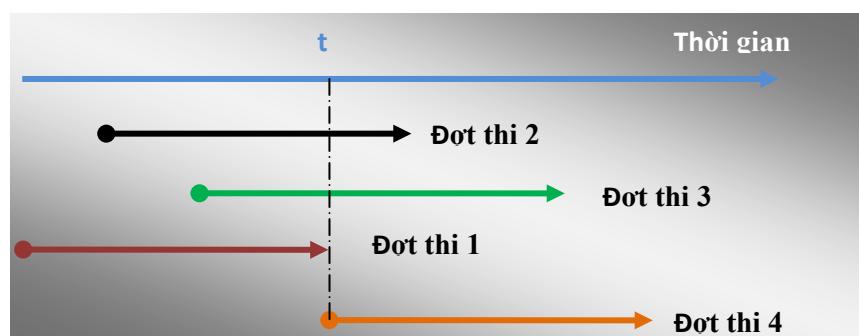
**Hình 13. Quy trình cấp chứng chỉ**

## 1.2. Các vấn đề còn tồn tại

Hiện tại, việc thực hiện các công việc trong quy trình thi được giám sát và điều phối một cách thủ công. Khi mỗi đợt thi đến gần, nhân viên quản lý sẽ phân công các nhân viên thực hiện việc ghi danh thí sinh dự thi, lập danh sách thi, phân công coi thi v.v... Sau đó, trong quá trình thực hiện các công việc, người quản lý sẽ phải theo dõi, đôn đốc các công việc nhằm đảm bảo cho quy trình tổ chức thi được diễn ra đúng kế hoạch đã định.

Phương pháp quản lý một cách thủ công chiếm nhiều thời gian, đồng thời cũng có nhiều hạn chế trong việc phát hiện và xử lý các tình huống gây trễ hạn công việc. Để theo dõi tiến độ, người quản lý phải giữ liên lạc thường xuyên với các nhân viên của mình để yêu cầu cung cấp thông tin về tình hình công việc. Để có được một cái nhìn tổng thể về ứng dụng và đưa ra các nhận định, dự báo... người quản lý phải tập hợp các thông tin đã thu thập được rồi thực hiện thống kê, so sánh, mô hình hóa... Ngoài ra, tại một thời điểm bất kỳ, có thể có nhiều quy trình thi cùng diễn ra đồng thời như mô tả ở hình vẽ H.4. Điều này làm cho việc quản lý cũng như việc thực hiện nhiệm vụ đối với các nhân viên khác trở nên khó khăn và nhiều áp lực hơn. Theo đó, có thể xảy ra các tình huống bất lợi sau:

- Nhân viên có thể nhầm lẫn về thời gian hoàn thành các công việc của các quy trình.
- Có thể xảy ra những thời điểm mà nhân viên thực hiện không xuể các công việc nếu như không được quản lý tốt.
- Khó khăn trong công tác quản lý, theo dõi tiến độ, dự báo, cảnh báo.



**Hình 14. Nhiều đợt thi có thể diễn ra song song**

## 2. Yêu cầu chức năng

Mục này đưa ra một số các chức năng chính được đề nghị cho một ứng dụng quản lý quy trình thi tổng quát. Bao gồm các chức năng phục vụ quá trình thực hiện

công việc của nhân viên thửa hành và chức năng phục vụ công tác quản lý của nhân viên quản lý.

### **2.1. Chức năng thông báo.**

Ứng dụng cần cung cấp chức năng thông báo cho nhân viên sử dụng chương trình dưới dạng email, tin nhắn... trước và trong các mốc thời gian quan trọng nhằm giúp nhân viên sắp xếp thời gian biểu của riêng mình và thực hiện công việc sớm nhất có thể. Cụ thể hơn, ứng dụng sẽ gửi thông báo trong các trường hợp sau:

- Với mỗi đợt thi, trước khi một công việc được bắt đầu hay kết thúc theo hạn định, ứng dụng tự động gửi thông báo tới cho nhân viên được phân công phụ trách công việc đó. Việc thông báo này là nhằm giúp gợi nhớ cho các nhân viên thực hiện đúng hạn trách nhiệm của mình, tránh xảy ra các tình trạng trễ hạn hoặc ứ đọng các công việc.
- VỚI NHỮNG CÔNG VIỆC CÓ KHẢ NĂNG TRỄ HẠN CAO (ví dụ: gần hết thời hạn mà công việc vẫn chưa được tiến hành,...), ứng dụng cũng phải tự động gửi thông báo nhắc nhở một lần nữa.
- VỚI NHỮNG CÔNG VIỆC HOÀN THÀNH SỚM HẠN<sup>37</sup>, ứng dụng sẽ tự động gửi thông báo đến người nhân viên chịu trách nhiệm cho công việc tiếp theo, giúp họ nhanh chóng sắp xếp và thực hiện ngay công việc này.

### **2.2. Chức năng theo dõi công việc**

Đây là một trong những chức năng quan trọng mà ứng dụng quản lý LCV phải có. Chức năng này cho phép nhân viên và người quản lý theo dõi LCV và tình trạng của từng công việc thành phần trên LCV đó. Cụ thể là:

#### **2.2.1. ĐỐI VỚI CÁC NHÂN VIÊN THỰC HIỆN CÁC CÔNG VIỆC:**

---

<sup>37</sup> Sớm hạn: Công việc hoàn thành trước thời gian dự kiến

Đối với các nhân viên thực hiện các công việc, sau khi đăng nhập, có thể quan sát được tiến độ của các công việc mà họ đang thực hiện. Ví dụ nhân viên giáo vụ có thể xem được các thông tin về tổng số bài thi, số bài được chấm, thời gian chấm được cho phép, phần trăm công việc đã hoàn thành... đối với từng đợt thi cụ thể mà nhân viên này được phân công.

Ngoài ra, các công việc mà nhân viên phải thực hiện trong tương lai gần cũng được hiển thị một cách rõ ràng giúp nhân viên có thể lên lịch làm việc sắp tới cho hiệu quả.

### **2.2.2. Đối với người quản lý:**

Đối với người quản lý, ứng dụng cần cung cấp chức năng cho phép theo dõi tiến độ của công việc trên toàn bộ LCV, bao gồm các thông tin: ngày bắt đầu và kết thúc theo quy định, ngày bắt đầu và kết thúc trong thực tế, người thực hiện công việc, v.v... Ngoài ra, với các công việc đang diễn ra, thông tin về tiến độ thực hiện công việc là rất quan trọng và cần phải theo dõi trong chức năng này.

## **2.3. Chức năng cập nhật thông tin**

Song song với việc theo dõi tiến độ thực hiện các công việc, nhân viên có nhu cầu cập nhật trạng thái của các công việc mà họ phụ trách. Bao gồm các thông tin sau:

- Thời gian bắt đầu và kết thúc công việc thực tế: Mỗi công việc trên LCV đều có thời điểm bắt đầu và kết thúc, hai mốc thời gian này xác định khoảng thời gian thực tế mà công việc được tiến hành.
- Khối lượng công việc hoàn thành: là mức độ hoàn thành công việc đối với một công việc đang được tiến hành. Thông tin này được cung cấp bởi chính nhân viên thực hiện công việc đó.

- Thông tin ghi chú: Trong các trường hợp các công việc được hoàn thành sớm hoặc trễ hơn so với khoảng thời gian quy định, người quản lý cần biết nguyên nhân gây ra việc sớm hoặc trễ hạn đó. Ngoài ra, các thông tin khác mà bản thân người thực hiện công việc cho là quan trọng cũng có thể được lưu trữ dưới dạng này.

Ngoài ra, sau khi hoàn thành một công việc, nhân viên cần phải cung cấp các đơn từ, biểu mẫu (nếu có) liên quan đến công việc này cho ứng dụng. Những dữ liệu này vừa có tác dụng chứng minh rằng công việc đã thực sự kết thúc, vừa là cách để cung cấp các kết quả công việc cho người quản lý phục vụ cho công tác quản lý và cải tiến quy trình về sau. Do vậy ứng dụng cần phải linh hoạt trong quá trình tương tác với người dùng, tiếp nhận và lưu trữ hợp lý các thông tin đó.

Ứng dụng cung cấp chức năng thống kê từ tổng quát đến chi tiết nhằm giúp người quản lý có cái nhìn toàn diện về quy trình. Cụ thể như sau:

## **2.4. Chức năng thống kê**

### **2.4.1. Thông kê tổng thể:**

Dựa trên dữ liệu về các đợt thi đã thực hiện trong suốt quá trình từ khi ứng dụng hoạt động đến thời điểm hiện tại, từ đó đưa ra các thống kê tổng thể theo thời gian theo các tiêu chí khác nhau: số đợt thi hoàn thành sớm hạn, số đợt thi bị trễ hạn, tỉ lệ hoàn thành của một công đoạn cụ thể đối với các đợt thi... Các tiêu chí này được xây dựng sẵn trên ứng dụng, đồng thời cho phép người quản lý có thể tạo thêm các tiêu chí thống kê mới theo nhu cầu của thực tế nghiệp vụ.

### **2.4.2. Thông kê chi tiết:**

Đối với những đợt thi đã thực hiện xong, người quản lý có thể có nhu cầu xem các thông tin chi tiết: những công việc nào bị trễ hạn, những công việc nào được

hoàn thành sớm hạn, nguyên nhân gây sớm/trễ hạn, những nhân viên được phân công thực hiện công việc...

#### **2.4.3. Thông kê so sánh:**

So sánh các đợt thi trong cùng khoảng thời gian nhưng khác niên khóa. Ví dụ, người quản lý có nhu cầu so sánh lượng thí sinh dự thi và tình hình thực hiện quy trình thi trong tháng 5 của năm 2009, 2008, 2007... Từ đó đưa ra dự đoán về tình hình tháng 5, năm 2010, 2011... để có chuẩn bị cần thiết...

#### **2.4.4. Thông kê theo thời gian:**

Ứng dụng cần cung cấp chức năng thống kê theo thời gian. Cụ thể, chức năng này cho biết trong một khoảng thời gian hiện tại có những đợt thi nào diễn ra, đợt thi nào đã hoàn thành, chưa hoàn thành, đợt thi nào có nguy cơ bị trễ hạn; đối với mỗi đợt thi, ứng dụng cũng sẽ cung cấp chức năng thống kê cụ thể tình hình hoạt động của các công việc trong đợt thi đó ... Kết quả thống kê dạng này được hiển thị dưới dạng sơ đồ Gantt.

### **2.5. Chức năng cảnh báo:**

Tương tự như chức năng thông báo, đối với những công việc có nguy cơ trễ hạn cao, ứng dụng sẽ kích hoạt chức năng cảnh báo. Chức năng này được biểu hiện ở hai dạng:

- Gửi thông báo đến các nhân viên có liên quan nhằm nhắc nhở nhân viên nhanh chóng thực hiện công việc và đến người quản lý với thông tin về phần trăm công việc và người chịu trách nhiệm nhằm giúp quản lý nhanh chóng kiểm soát tình hình và đưa ra giải quyết phù hợp.
- Hiển thị dưới dạng màu sắc trên lược đồ của người quản lý cùng các thông tin cần thiết (phần trăm đã thực hiện được, người chịu trách nhiệm...) để

nhân viên quản lý có thể nhanh chóng phát hiện phạm vi được cảnh báo (công việc gì) cùng những thông tin đó để đưa ra tình huống xử lý thích hợp.

## **2.6. Chức năng phân quyền:**

Ứng dụng phải có chức năng cho phép phân công các nhân viên phụ trách các công việc. Sự phân công này cần phải có khả năng thay đổi khi cần thiết. Bao gồm các dạng sau:

Phân công mặc định: Quy trình thi được diễn ra lặp lại nhiều năm. Bộ phận tổ chức thi bao gồm các nhân viên của trung tâm được phân công nắm giữ các công việc một cách ổn định trong thời gian dài. Do vậy, mỗi khi mở một đợt thi mới nhân viên quản lý không cần thiết thực hiện tao tác phân công, thay vào đó ứng dụng phải có khả năng tự động ghi nhận phân công.

Phân công lại: Trải qua nhiều năm làm việc, có thể sẽ có nhiều tình huống cần thực hiện phân công lại các công việc cho nhân viên (thay đổi quy trình, thay đổi nhân sự...). Do đó ứng dụng cũng phải cung cấp khả năng thực hiện phân công lại khi cần.

Phân công trong một khoảng thời gian: Chức năng này cho phép phân công công việc trong một khoảng thời gian giới hạn. Sau khoảng thời gian này người quản lý cần phải thực hiện gia hạn hoặc phân công lại cho công việc đó. Chức năng này hữu ích trong trường hợp có nhân viên A được cử đi công tác một thời gian, người quản lý muốn phân công nhân viên B đảm nhận tạm thời công việc trong thời gian A đi công tác, sau đó sẽ trở lại bình thường.

Phân công riêng (phân công cụ thể): Chức năng này cho phép phân công công việc trên một đợt thi riêng nào đó. Chức năng này đặc biệt hữu ích đối với những đợt thi có số lượng thí sinh đột biến<sup>38</sup>, cần được sắp xếp lại chế độ phân công cho hợp lý nhằm đảm bảo công việc được hoàn thành đúng hạn với hiệu quả cao nhất có thể.

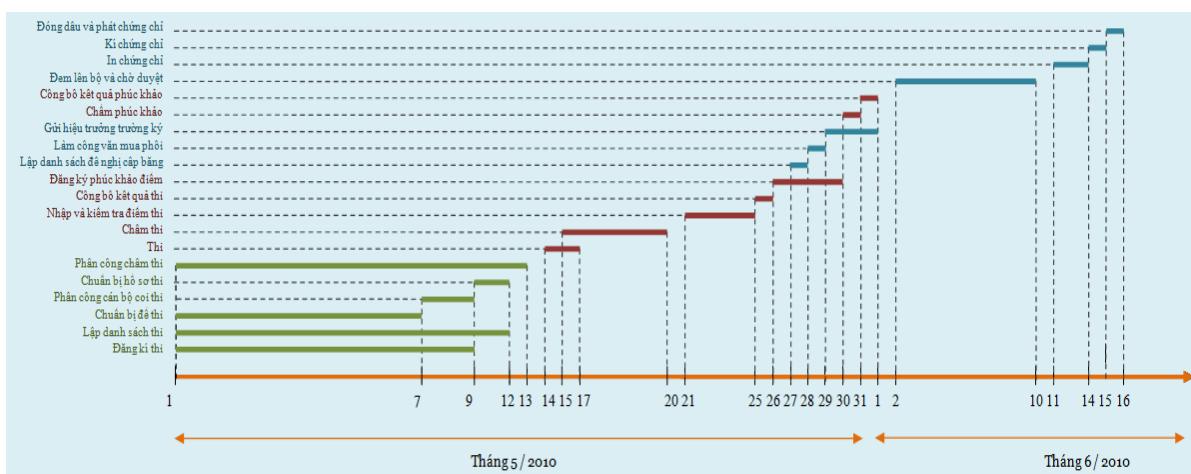
## **2.7. Chức năng biểu diễn luồng công việc dưới dạng sơ đồ trực quan**

Để tăng tính tiện dụng và thân thiện, ứng dụng phải cung cấp chức năng biểu diễn LCV dưới dạng sơ đồ nhằm giúp người quản lý có cái nhìn trực quan hơn đối với quy trình đang thực thi. Các dạng sơ đồ đề nghị là sơ đồ Gantt, sơ đồ LCV,...

---

<sup>38</sup> Nhiều hơn hoặc ít hơn rất nhiều so với các đợt thi khác, đòi hỏi người quản lý phải thêm hoặc bớt nhân viên thực hiện các công đoạn trong đợt thi

Các thông tin liên quan đến các đợt thi được bố trí một cách hợp lý sao cho mang lại hiệu suất làm việc tốt nhất cho người dùng. Hình H5 là một ví dụ cho cách biểu diễn dưới dạng sơ đồ Gantt các công việc của một đợt thi diễn hành.



**Hình 15. Biểu diễn quy trình dưới dạng sơ đồ Gantt**

### 3. Yêu cầu phi chức năng

#### 3.1. Tính tiên hóa

Ứng dụng cần có tính tiên hóa, cho phép có khả năng nâng cấp ứng dụng trong tương lai mà không ảnh hưởng đến các chức năng đang được sử dụng. Các nâng cấp có thể xảy ra như: thay đổi hệ quản trị cơ sở dữ liệu, thay đổi thời gian thực hiện quy trình, thay đổi nhân sự,...

#### 3.2. Yêu cầu về giao diện

- Giao diện đồ họa dễ sử dụng, trực quan, thân thiện với nhân viên.
- Hỗ trợ chức năng thống kê, theo dõi luồng công việc dạng sơ đồ quen thuộc với người quản lý.
- Ứng dụng có hướng dẫn đầy đủ, ít tương tác, hỗ trợ tối đa cho người sử dụng.

### 3.3. Tính hiệu quả

- Các thao tác cung cấp dữ liệu đầu vào cho luồng công việc gọn nhẹ, đơn giản, không phức tạp.
- Cập nhật nhanh, hỗ trợ nhiều nhân viên có thể cùng truy cập ứng dụng cùng lúc mà không bị hiện tượng thắt cổ chai.

## 4. Phân tích mức quan niệm

Đến đây, chúng ta đã biết được các tính chất quan trọng của mô hình LCV cũng như các yêu cầu cụ thể đối với một ứng dụng quản lý LCV là như thế nào. Tùy vào từng lĩnh vực khác nhau, mỗi hệ thống ứng dụng quản lý được xây dựng cho lĩnh vực đó sẽ có sự khác biệt về các yêu cầu nghiệp vụ. Ứng với mỗi quy trình nghiệp vụ chúng ta phải MHH thành một sơ đồ LCV sao cho thỏa mãn tất cả các yêu cầu logic của nghiệp vụ đó, đồng thời phải có khả năng tương tác với người dùng trong các chức năng điều khiển hoạt động thực thi công việc.

Ngoài ra, chúng ta cũng đã biết được các chức năng quan trọng cần có ở một ứng dụng quản lý LCV tổ chức thi gồm những gì, mỗi một chức năng ấy có vai trò và vị trí như thế nào v.v.... Trong phần này, chúng tôi sẽ thể hiện lại kết quả phân tích và thiết kế ứng dụng trên.

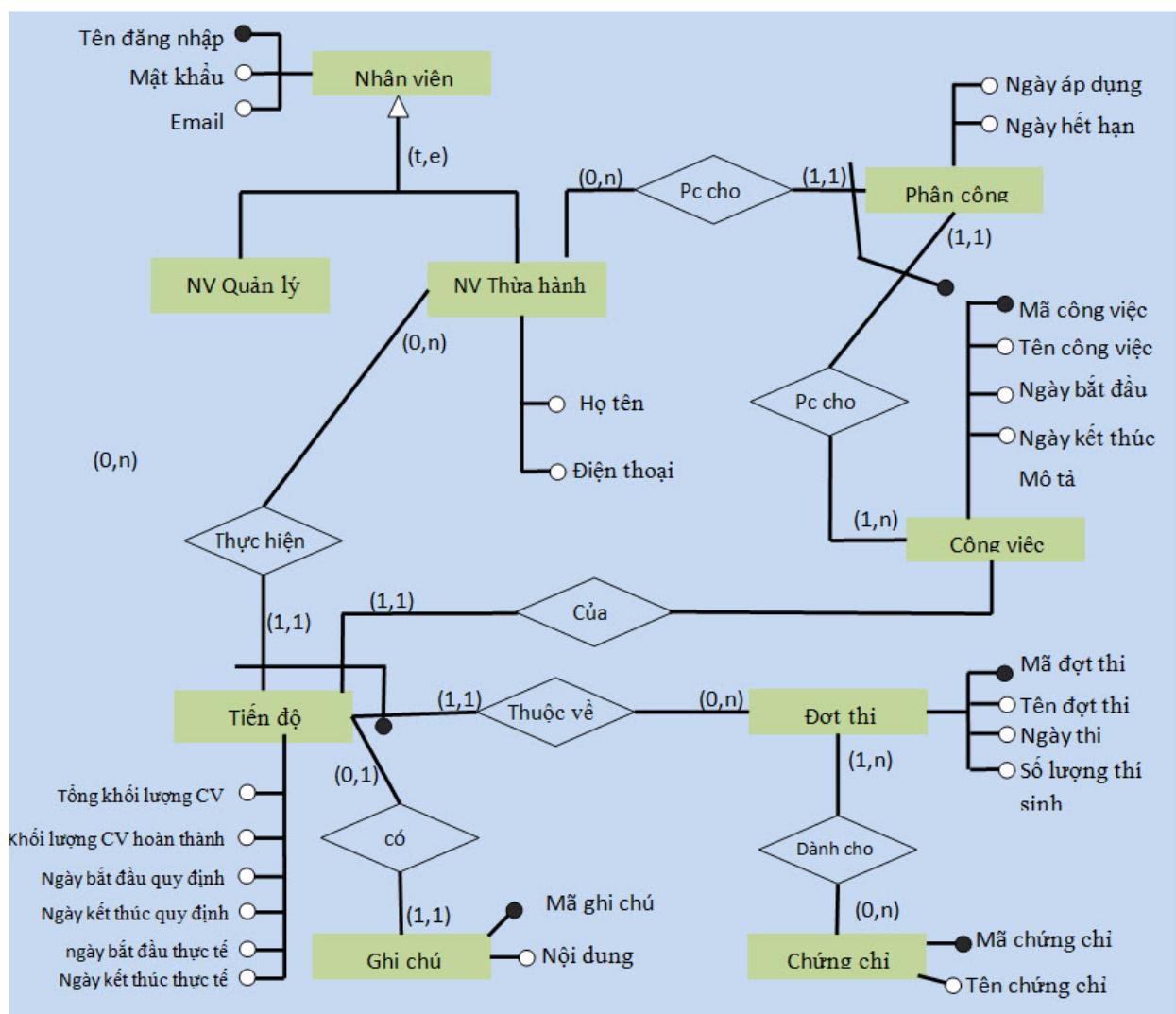
### 4.1. Mô hình dữ liệu ở mức quan niệm

Phần này mô tả chi tiết về mô hình dữ liệu ở mức quan niệm. Các thực thể (đối tượng dữ liệu) mà ứng dụng cần phải lưu trữ bao gồm:

STT	Tên	Mô tả
1	<b>Nhân viên</b>	Bao gồm nhân viên quản lý và nhân viên thừa hành
2	<b>Công việc</b>	Các công việc trong mỗi đợt thi

<b>3</b>	<b>Phân công</b>	Sự phân công công việc cho các nhân viên
<b>4</b>	<b>Đợt thi</b>	Các đợt thi cấp chứng chỉ hằng năm
<b>5</b>	<b>Tiến độ</b>	Tiến độ của mỗi sự phân công, tương ứng với từng đợt thi
<b>6</b>	<b>Ghi chú</b>	Thông tin ghi chú cho quá trình thực hiện công việc
<b>7</b>	<b>Chứng chỉ</b>	Các chứng chỉ được tổ chức thi trong mỗi đợt thi

**Bảng 31. Các đối tượng dữ liệu được lưu trữ**



**Hình 16. Mô hình dữ liệu quan niệm**

Ghi chú:

Ký hiệu	Ý nghĩa
	Thực thể. Là một đối tượng dữ liệu của hệ thống
	Mối kết hợp giữa các thực thể
	Thuộc tính
	Thuộc tính khóa
	Khóa của thực thể A là tập hợp bao gồm x và y. Trong đó x,y lần lượt là thuộc tính khóa của B và C.

**Bảng 32. Ý nghĩa kí hiệu trong mô hình dữ liệu mức quan niệm**

Hình H8 thể hiện lược đồ dữ liệu ở mức quan niệm. Bao gồm các thực thể, thuộc tính cho mỗi thực thể và mối quan hệ giữa chúng.

Thành phần nhân viên thuộc bộ phận tổ chức thi của một đơn vị đào tạo chứng chỉ quốc gia thông thường bao gồm hai loại nhân viên cơ bản, đó là nhân viên quản lý và nhân viên thửa hành. Nhân viên thửa hành là người trực tiếp chịu trách nhiệm thực hiện các công việc được phân công bởi nhân viên quản lý. Ngoài các thông tin chung tối thiểu để sử dụng ứng dụng, nhân viên thửa hành còn phải cung cấp thêm họ tên đầy đủ và số điện thoại liên lạc dùng trong công tác quản lý của người quản lý.

Thực thể Công việc lưu trữ các công việc thành phần trong một đợt thi (ghi danh, lập danh sách, coi thi...). Thông tin ngày bắt đầu và ngày kết thúc của thực thể Công việc là ngày thứ tự (lưu trữ dưới dạng số nguyên) lấy ngày tổ chức thi làm chuẩn<sup>39</sup>.

<sup>39</sup> Một công việc có giá trị ngày bắt đầu là n nghĩa là nó được bắt đầu sau n ngày so với ngày tổ chức thi. Ngược lại, ngày bắt đầu nhận giá trị -m mang ý nghĩa công việc phải được bắt đầu trước m ngày so với ngày thi

Mỗi công việc sẽ được phân công cho một hoặc nhiều nhân viên cùng phụ trách. Đồng thời, một nhân viên nào đó có thể được phân công một hoặc nhiều công việc khác nhau. Để phục vụ lưu trữ cho chức năng phân công mặc định như đã mô tả ở phần yêu cầu của khách hàng, mỗi thẻ hiện phân công được lưu trữ cố định tương ứng với một nhân viên thửa hành và một công việc. Ngoài ra, mỗi thẻ hiện phân công được kèm theo thông tin ngày áp dụng và ngày hết hạn phân công đó. Trong trường hợp sự phân công là không có thời hạn, thuộc tính ngày hết hạn mang giá trị null.

Trên thực tế, một đợt thi có thể được tổ chức cho một hoặc nhiều loại chứng chỉ. Và tương ứng với các đợt thi khác nhau, mỗi một sự phân công sẽ có quá trình thực thi khác nhau. Thực thể Tiến độ chịu trách nhiệm theo dõi quá trình thực thi đó. Các mốc thời gian ngày bắt đầu, ngày kết thúc của mỗi công việc sẽ được chuyển đổi ra ngày lịch, lưu trữ ở hai thuộc tính Ngày bắt đầu quy định và Ngày kết thúc quy định.

Cuối cùng, trong một số trường hợp công việc được kết thúc trễ hạn hoặc sớm hạn so với quy định, có thể nhân viên quản lý sẽ cần biết nguyên nhân gây ra sự sớm/trễ hạn đó. Để phục vụ chức năng này, mỗi thẻ hiện Tiến độ được gắn kết với tối đa một thẻ hiện Ghi chú (Có thể có hoặc không có). Ngược lại, một thông tin ghi chú luôn luôn dành cho duy nhất một tiến độ.

Dưới đây sẽ liệt kê thông tin chi tiết các thuộc tính của từng thực thể bao gồm tên gọi, miền giá trị, ý nghĩa,...

#### **4.1.1 Nhân viên**

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Tên đăng	Chuỗi	50	K	Tên đăng nhập vào ứng

	nhập				dụng.
2	Mật khẩu	Chuỗi	250	K	Mật khẩu đăng nhập
3	Email	Chuỗi	250	K	Địa chỉ email.

#### 4.1.2 Nhân viên quản lý & Nhân viên thừa hành

Nhân viên quản lý và nhân viên thừa hành đều thừa kế các thuộc tính từ thực thể nhân viên. Riêng nhân viên thừa hành cần có thêm các thông tin sau:

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Họ tên	Chuỗi	50	K	Tên đầy đủ của nhân viên thừa hành
2	Điện thoại	Chuỗi	11	C	Điện thoại liên lạc

**Bảng 34.Thông tin chi tiết thực thể Nhân viên thừa hành**

#### 4.1.3 Công việc

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Mã công việc	Số nguyên dương		K	Mỗi công việc được phân biệt bởi một mã duy nhất.
2	Tên công việc	Chuỗi	100	K	Tên công việc
3	Ngày bắt đầu	Số nguyên		K	Ngày bắt đầu công việc theo quy định
4	Ngày kết thúc.	Số nguyên		K	Ngày kết thúc công việc theo quy định
5	Mô tả	Chuỗi	250	K	Mô tả công việc

**Bảng 35.Thông tin chi tiết thực thể Công việc**

#### 4.1.4 Phân công

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Ngày áp dụng	ngày tháng		K	Ngày đầu tiên áp dụng phân công
2	Ngày hết hạn	Ngày tháng		K	Ngày mà sự phân công tương ứng hết hiệu lực.

Bảng 36.Thông tin chi tiết thực thể Phân Công

#### 4.1.5 Đợt thi

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Mã đợt thi	Số nguyên		K	Mã đại diện cho một đợt thi.
2	Tên đợt thi	Chuỗi	250	K	Tên đợt thi
3	Ngày thi	Ngày tháng		K	Ngày tổ chức thi
4	Số lượng thí sinh	Số nguyên dương		K	Số lượng thí sinh dự kiến tham dự kỳ thi.

Bảng 37.Thông tin chi tiết thực thể Đợt thi

#### 4.1.6 Tiết độ

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Tổng khối lượng CV	Số nguyên dương		C	Tổng khối lượng công việc cần phải hoàn thành
2	Khối lượng CV hoàn thành	Số nguyên dương		C	Khối lượng công việc đã hoàn thành
3	Ngày bắt đầu quy định	Ngày tháng		K	Ngày bắt đầu công việc theo quy định (ngày lịch)
4	Ngày kết thúc quy định	Ngày tháng		K	Ngày kết thúc công việc theo quy định (ngày lịch)
5	Ngày bắt đầu thực tế	Ngày tháng		C	Ngày công việc thực sự được nhân viên thừa hành bắt đầu
6	Ngày kết thúc thực tế	Ngày tháng		C	Ngày công việc thực sự được nhân viên thừa hành kết thúc.

Bảng 38. Thông tin chi tiết thực thể tiến độ

#### 4.1.7 Ghi chú

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Mã ghi chú	Số nguyên		K	Khóa chính, là mã duy nhất cho mỗi ghi chú.
2	Nội dung	Văn bản	1024	K	Nội dung ghi chú

Bảng 39. Thông tin chi tiết thực thể Ghi chú

#### 4.1.8 Chứng chỉ

STT	Tên thuộc tính	Miền giá trị	Kích thước	Cho phép null (C/K)	Ý nghĩa
1	Mã chứng chỉ	Số nguyên dương		K	Mã đại diện cho mỗi chứng chỉ
2	Tên chứng chỉ	Chuỗi	250	K	Tên chứng chỉ.

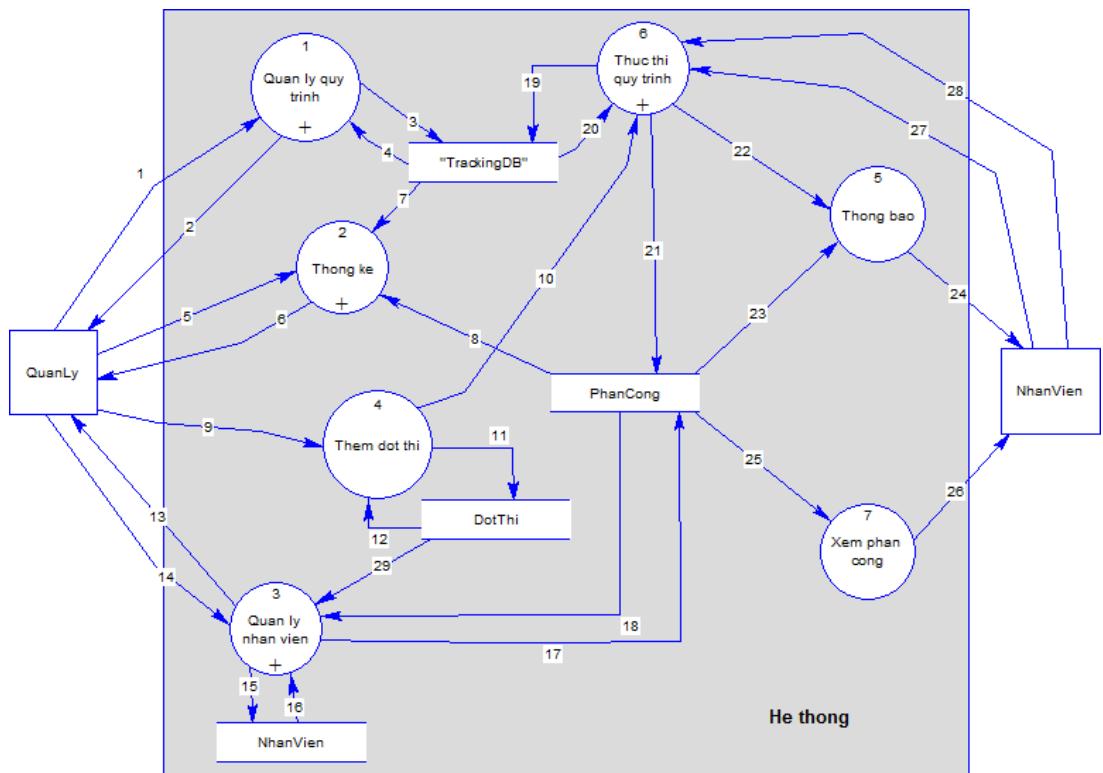
Bảng 40. Thông tin chi tiết thực thể chứng chỉ

#### 4.2. Mô hình xử lý mức quan niệm

Phần này mô hình hóa một cách chi tiết các xử lý của ứng dụng ở mức quan niệm. Bao gồm các cụm xử lý chính, tác nhân đầu/cuối và các luồng dữ liệu luân chuyển giữa các đối tượng.

##### 4.2.1. Tổng quan

Ở mức tổng quát, ứng dụng có 7 cụm xử lý chính (Hình vẽ H9.), cung cấp 7 chức năng chính của một ứng dụng quản lý luồng công việc thi, cấp chứng chỉ và in văn bằng nói chung. Đó là các xử lý sau:



**Hình 17. Mô hình xử lý tổng quát**

Giải thích các dòng dữ liệu lưu chuyển:

Số	Diễn giải	Số	Diễn giải
1	Yêu cầu quản lý quy trình	15	Thông tin nhân viên cập nhật
2	Kết quả quản lý quy trình	16	Thông tin nhân viên
3	Cấu trúc quy trình	17	Dữ liệu cập nhật phân công
4	Cấu trúc quy trình cập nhật	18	Dữ liệu phân công
5	Yêu cầu thống kê	19	Dữ liệu thực thi quy trình cập nhật
6	Kết quả thống kê	20	Dữ liệu thực thi quy trình
7	Dữ liệu thực thi quy trình	21	Tiến độ thực thi quy trình
8	Dữ liệu phân công	22	Mã thông báo
9	Ngày thi	23	Dữ liệu phân công
10	Yêu cầu tạo thể hiện	24	Nội dung thông báo
11	Dữ liệu đợt thi cập nhật	25	Dữ liệu phân công

12	Dữ liệu đợt thi	26	Lịch phân công dạng sơ đồ gantt
13	Kết quả thực hiện quản lý ND	27	Trạng thái công việc
14	Yêu cầu quản lý nhân viên	28	Thông tin ghi chú thực thi
		29	Ngày thi

**Bảng 41. Các dòng dữ liệu di chuyển trong mô hình xử lý tổng quát**

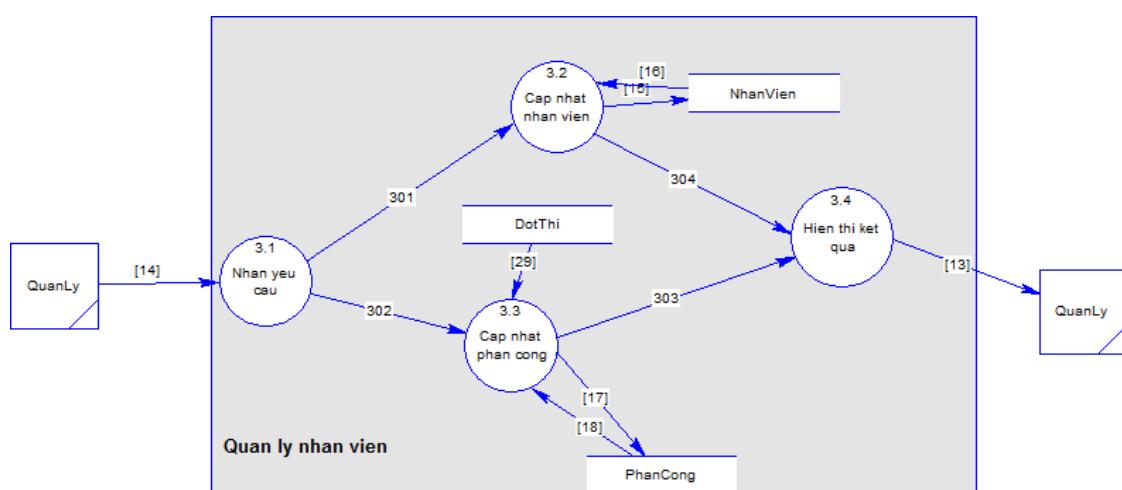
- Quản lý quy trình: xử lý này bao gồm các xử lý thành phần như xem cấu trúc của một luồng công việc, tiến hành các cập nhật đối với luồng công việc đó (thêm, xóa, sửa các công việc)... Xử lý này nhận vào các lệnh quản lý và trả ra kết quả sau khi thực hiện các lệnh quản lý đó.
- Thông kê: Bao gồm các xử lý thống kê quá trình thực thi luồng công việc đối với các đợt thi. Đầu vào là các lệnh yêu cầu thống kê, kết quả trả ra là các dạng sơ đồ thể hiện kết quả thống kê (sơ đồ gantt, biểu đồ, lược đồ mô phỏng luồng công việc ...)
- Thêm đợt thi: Xử lý này phục vụ cho chức năng thêm đợt thi mới. Khi cần thêm một đợt thi, người quản lý sẽ cung cấp cho ứng dụng ngày thi của đợt thi đó. Xử lý này sẽ tự động gửi yêu cầu tạo ra một luồng công việc mới, đồng thời khởi tạo các giá trị dùng cho việc quản lý đợt thi và lưu trữ xuống cơ sở dữ liệu.
- Quản lý nhân viên: thực hiện các xử lý cập nhật thông tin, cập nhật phân công công việc cho nhân viên.
- Thực thi quy trình: Bao gồm các xử lý cho phép nhân viên tương tác với ứng dụng để cập nhật trạng thái công việc mà họ đang thực hiện, cập nhật thông tin ghi chú thực thi nếu có...
- Thông báo: Trong quá trình thực thi quy trình thi, ứng dụng sẽ tự động phát sinh các thông báo cho nhân viên được phân công công việc nếu như công việc đó sắp được bắt đầu, sắp kết thúc, có nguy cơ trễ hạn ... Các yêu cầu

thông báo đó sẽ được đơn vị xử lý “ThongBao” tiếp nhận và phát sinh các nội dung thông báo phù hợp để gửi tới nhân viên phụ trách công việc.

- Xem phân công: Mỗi nhân viên khi đăng nhập vào ứng dụng có quyền xem lịch công tác mà họ được phân công trong từng đợt thi. Đơn vị xử lý này sẽ đọc dữ liệu phân công từ cơ sở dữ liệu và hiển thị dưới dạng sơ đồ gantt.

#### 4.2.2. Quản lý nhân viên

Cụm quản lý nhân viên bao gồm các xử lý thêm/xóa/sửa một nhân viên ứng dụng và thêm/xóa/sửa các phân công công việc cho nhân viên đó.

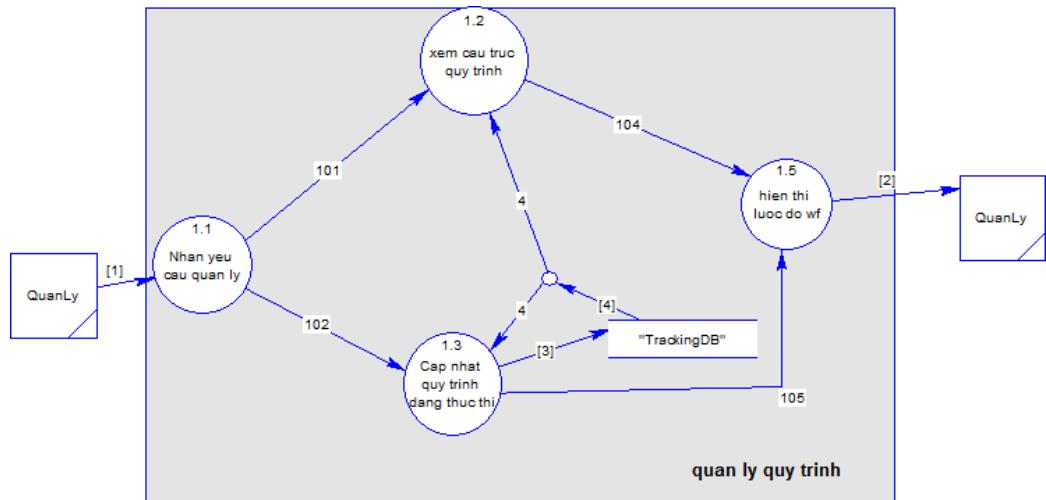


**Hình 18. Mô hình xử lý Quản lý nhân viên**

Số	Diễn giải	Số	Diễn giải
301	thông tin nhân viên cần cập nhật	304	Kết quả cập nhật nhân viên
302	thông tin phân công cần cập nhật	305	Kết quả cập nhật phân công

**Bảng 42. Các dòng dữ liệu di chuyển trong mô hình xử lý quản lý nhân viên**

#### 4.2.3. Quản lý quy trình

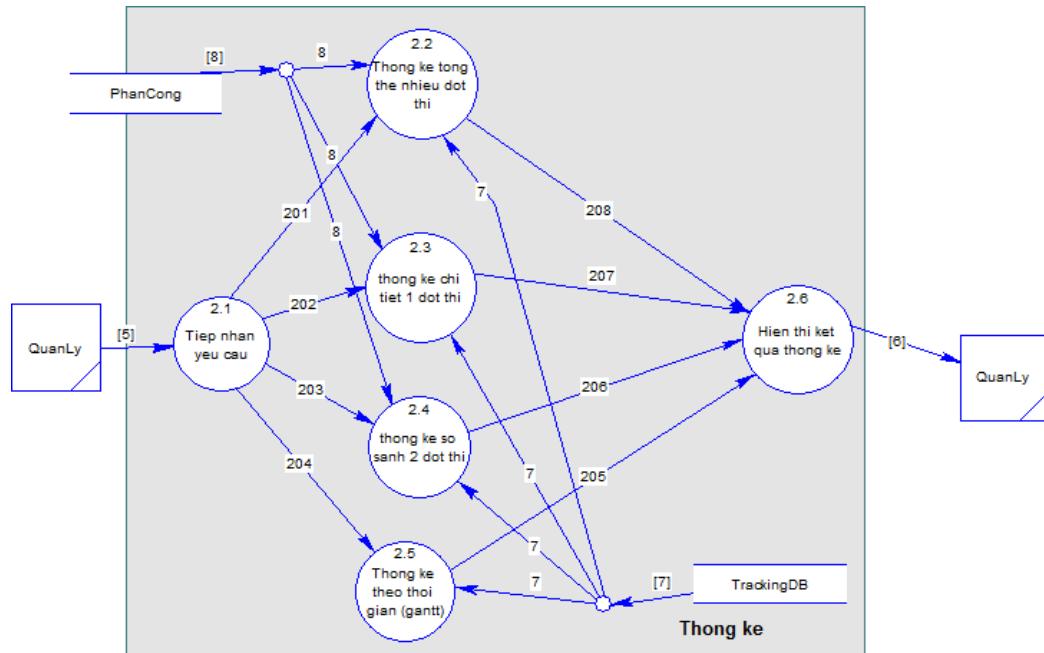


**Hình 19. Mô hình xử lý Quản lý quy trình**

#### 4.2.4. Thống kê

Cụm xử lý thống kê bao gồm 4 loại thống kê sau:

- Thống kê tổng quát: tiến hành thống kê quá trình thực thi cho các đợt thi đã kết thúc. Bao gồm các thống kê về số lượng thí sinh dự thi qua các đợt, tỉ lệ đúng hạn, sớm và trễ hạn...
- Thống kê chi tiết một đợt thi: thống kê ở cấp độ chi tiết các công việc trong từng đợt thi cụ thể. Những công việc đúng/sớm/trễ hạn, các thông tin ghi chú cho từng công việc (nếu có),...
- Thống kê so sánh: so sánh việc thực thi các công việc giữa hai đợt thi bất kỳ.
- Thống kê theo thời gian: thống kê trong một khoảng thời gian nhất định có bao nhiêu đợt thi đang diễn ra, tiến độ của từng đợt thi... kết quả của loại thống kê này được biểu diễn dưới dạng sơ đồ Gantt.



**Hình 20. Mô hình xử lý Thông kê**

Số	Diễn giải	Số	Diễn giải
201	Yêu cầu thống kê tổng quát	205	Kết quả thống kê tổng quát
202	Yêu cầu thống kê 1 đợt thi	206	Kết quả thống kê 1 đợt thi
203	Yêu cầu thống kê so sánh	207	Kết quả thống kê so sánh
204	Yêu cầu thống kê theo thời gian	208	Kết quả thống kê theo thời gian

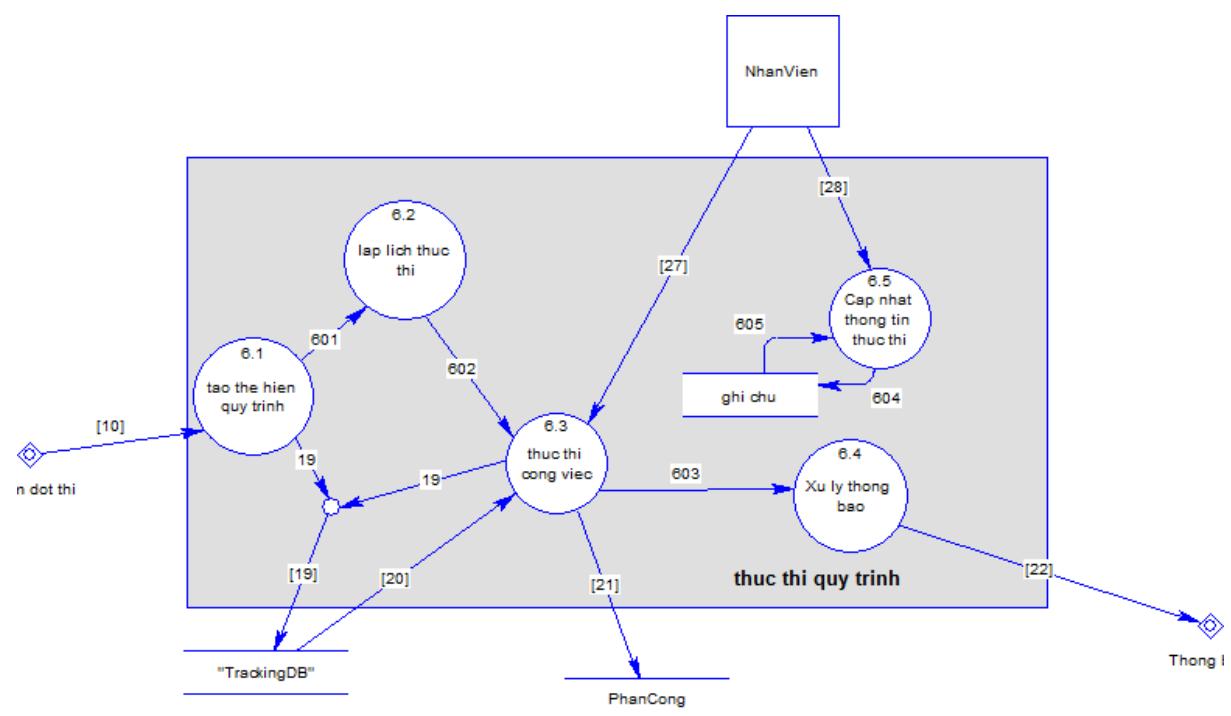
**Bảng 43. Các dòng dữ liệu di chuyển trong mô hình xử lý thống kê**

#### 4.2.5. Thực thi quy trình

- Tạo thẻ hiện quy trình mới: tạo một đối tượng LCV cho đợt thi mới.
- Lập lịch thực thi: Mỗi đợt thi đặc trưng bởi một ngày thi. Xử lý “lập lịch thực thi” chịu trách nhiệm tính toán các mốc thời gian thực tế tương ứng với thời hạn đã được mô tả trong quy trình.
- Thực thi activity: Nhận thông tin đầu vào từ nhân viên là trạng thái mới nhất về công việc mà họ được phân công thực hiện, xử lý này tiến hành cập nhật

vào ứng dụng trạng thái của công việc nhằm phục vụ cho công tác quản lý luồng công việc.

- Cập nhật thông tin thực thi: trong trường hợp công việc trễ hạn hoặc sớm hạn so với quy định, người quản lý có nhu cầu biết nguyên nhân gây nên sự sớm hoặc trễ hạn đó. Xử lý này nhận thông tin từ nhân viên và tiến hành cập nhật thông tin đó vào ứng dụng.
- Nhận các yêu cầu thông báo trong quá trình thực thi các công việc, tiến hành đối sánh và gửi mã thông báo phù hợp ra ngoài.



**Hình 21. Mô hình xử lý Thực thi quy trình**

Số	Diễn giải	Số	Diễn giải
601	Ngày thi	604	Thông tin ghi chú thực thi cập nhật
602	Lịch công việc	605	Thông tin ghi chú thực thi
603	Yêu cầu thông báo		

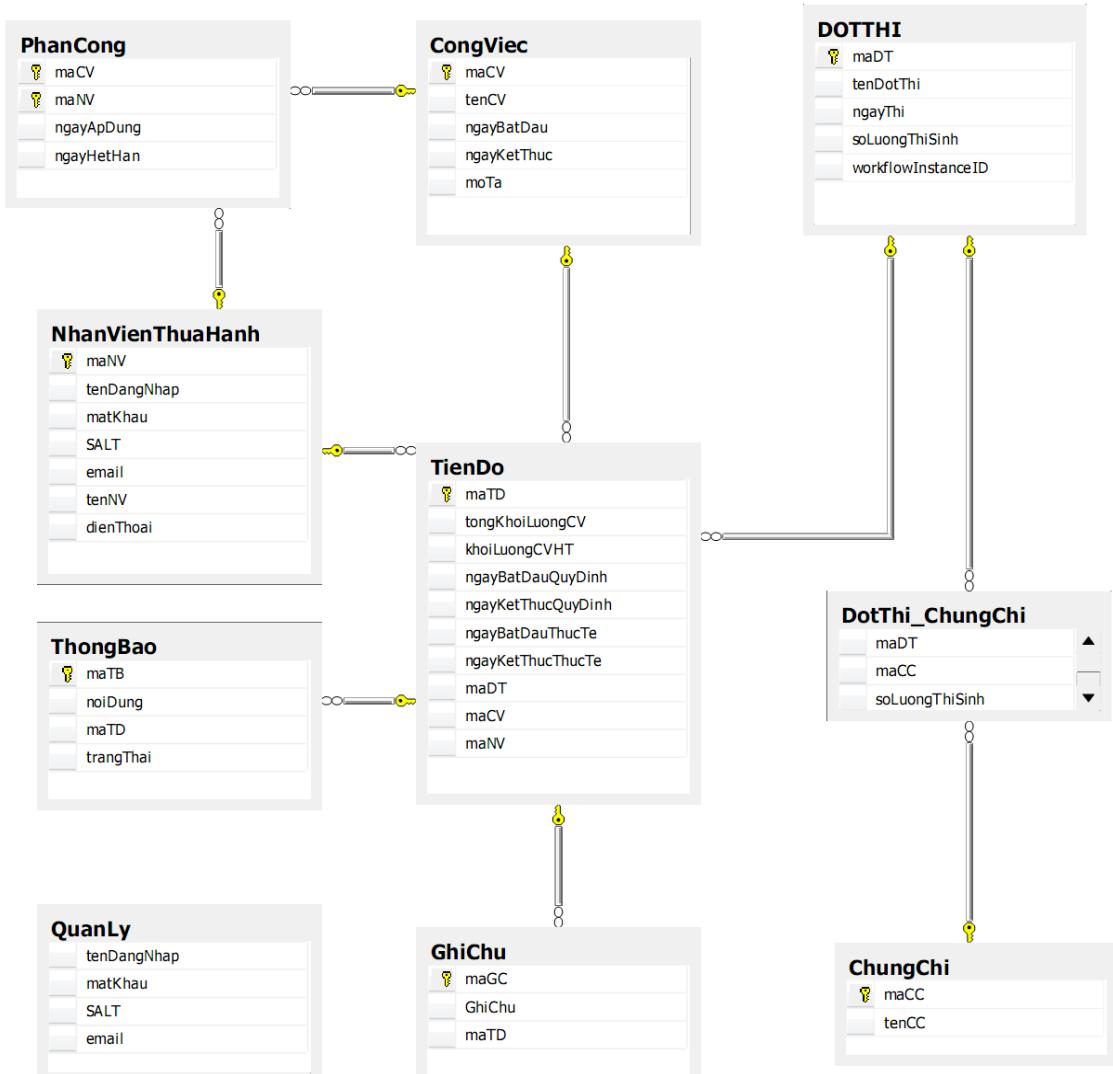
**Bảng 44. Các dòng dữ liệu di chuyển trong mô hình xử lý thực thi quy trình**

## 5. Cài đặt mức vật lý

Phần trên đây vừa trình bày mô hình dữ liệu ở mức phân tích và chi tiết các thực thể cần được tổ chức lưu trữ. Mô hình này chỉ mới dừng lại ở mục đích là làm sao thỏa mãn được nhu cầu lưu trữ để thực hiện các chức năng của ứng dụng chứ chưa xét đến tính tối ưu về mặc lưu trữ cũng như khai thác dữ liệu. Phần này tiếp tục trình bày về mô hình dữ liệu được cài đặt thực tế trong ứng dụng, theo đó cũng diễn giải một số điểm tối ưu hóa đã được áp dụng lên lược đồ phân tích.

Như đã trình bày ở trên, nhân viên thuộc bộ phận thi gồm có hai loại: Nhân viên quản lý và Nhân viên thừa hành. Một phương pháp đã được đề nghị là gộp chung hai loại nhân viên này vào cùng một bảng dữ liệu (table) và thêm thuộc tính LoaiNV (loại nhân viên) để phân biệt. Tuy nhiên, theo tính toán, chi phí dùng cho việc lưu trữ, khai thác, kiểm tra... trong quá trình khai thác dữ liệu cao hơn nhiều so với việc lưu riêng ra hai bảng dữ liệu. Giải pháp được lựa chọn là lưu trữ danh sách nhân viên thừa hành vào bảng dữ liệu NhanVienThuaHanh, bảng dữ liệu QuanLy lưu thông tin người quản lý (Chỉ có một thể hiện). Ngoài ra, dùng một thuộc tính dạng số nguyên (maNV) để làm khóa chính cho bảng dữ liệu nhân viên cũng là một bước tối ưu hóa.

Để phục vụ cho chức năng phân công, bảng TienDo và bảng PhanCong cùng tham chiếu tới CongViec và NhanVienThuaHanh. Lúc này, bảng dữ liệu PhanCong chỉ đóng vai trò là một bảng danh mục, lưu trữ các thể hiện phân công mẫu sẽ được áp dụng mỗi khi chức năng phân công tự động được kích hoạt.

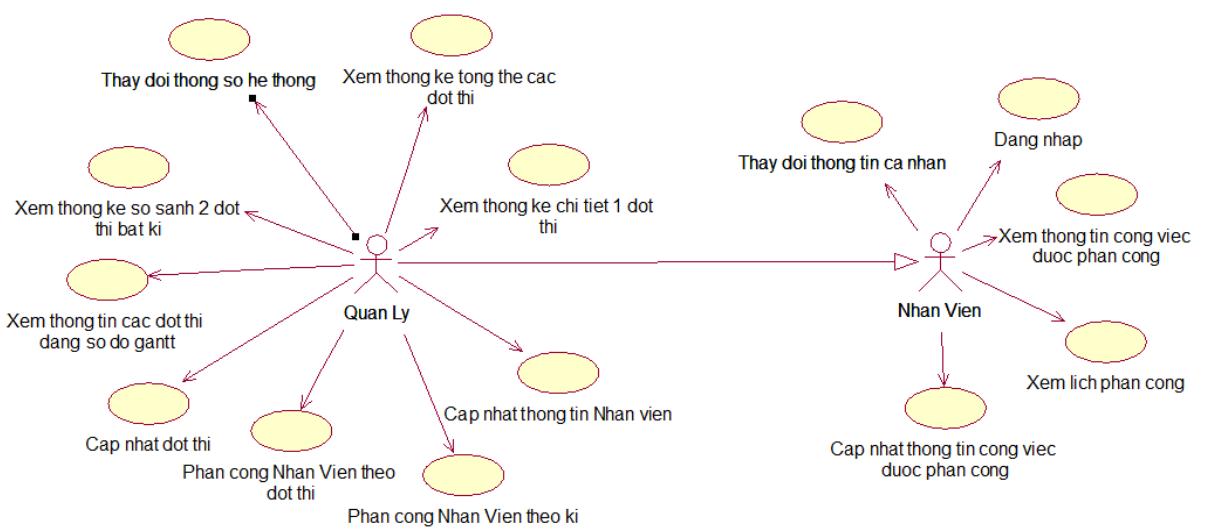


**Hình 22. Mô hình dữ liệu mức vật lý**

## 6. Mô tả chức năng và các giao diện

### 6.1. Thiết kế User-case

#### 6.1.1. Lược đồ User-case:



**Hình 23. Lược đồ User-Case**

#### 6.1.2. Danh sách các Actor

STT	Tên Actor	Ý nghĩa
1	Nhân viên	Các nhân viên có sử dụng đến ứng dụng để cung cấp thông tin thực thi các công đoạn trên luồng công việc
2	Quản lý	Người quản lý chính của luồng công việc

**Bảng 45. Danh sách các Actor**

#### 6.1.3. Danh sách các Use-case

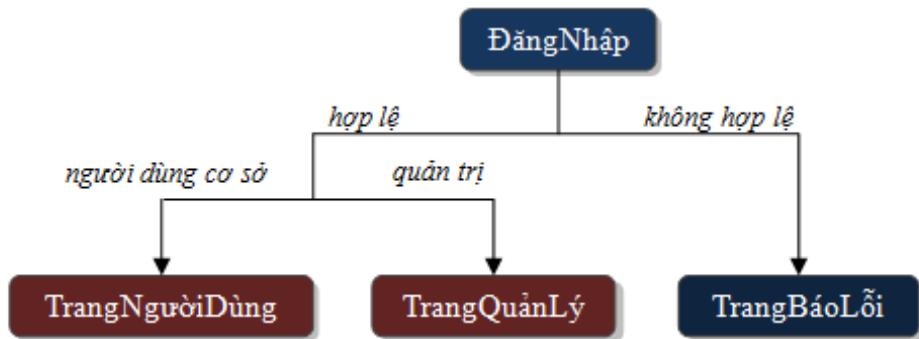
STT	Tên Use-case	Ý nghĩa
1	Đăng nhập	Đăng nhập sử dụng ứng dụng
2	Xem thông tin công việc được phân công	Xem thông tin về công việc thuộc trách nhiệm của nhân viên trên đợt thi
3	Xem lịch phân công	Xem lịch nhân viên được phân công thực hiện 1 công đoạn nào đó trên 1 đợt thi
4	Cập nhật thông tin công việc	Cập nhật thông tin về công việc thuộc

	được phân công	trách nhiệm của nhân viên trên đợt thi
5	Xem thống kê tổng thể các đợt thi	
6	Xem thống kê chi tiết 1 đợt thi	
7	Xem thống kê so sánh 2 đợt thi	
8	Xem thông tin các đợt thi dạng sơ đồ Gantt	
9	Phân công Nhân viên theo đợt thi	Phân công Nhân viên vào các công đoạn trên các đợt thi
10	Phân công Nhân viên theo kì	Phân công Nhân viên vào các công đoạn trong 1 khoảng thời gian cụ thể
11	Cập nhật thông tin Nhân Viên	Thay đổi (thêm, xóa, sửa) thông tin của nhân viên trên ứng dụng
12	Thay đổi thông tin cá nhân	
13	Thay đổi thông số ứng dụng	Cập nhật lại các thông số thời gian thực hiện các công đoạn tương ứng trong luồng công việc
14	Cập nhật đợt thi	Thay đổi (thêm, xóa, sửa) thông tin đợt thi

**Bảng 46. Danh sách các Use-case**

## 6.2. Thiết kế giao diện

### 6.2.1. Cấu trúc các màn hình xử lý chính



**Hình 24. Cấu trúc màn hình chính**

Khi nhân viên chọn sử dụng chương trình, đầu tiên, cửa sổ đăng nhập hiện ra và yêu cầu nhân viên cần phải đăng nhập trước khi có thể sử dụng chương trình. Nếu người đăng nhập với tài khoản hợp lệ (Tên tài khoản và Mật khẩu đúng với tên tài khoản và mật khẩu đã lưu trên cơ sở dữ liệu), cửa sổ đăng nhập sẽ đóng lại, và thay vào đó cửa sổ tương ứng với vai trò nhân viên sẽ hiện ra (nếu tài khoản phù hợp với loại nhân viên là Nhân viên, trang nhân viên sẽ được kích hoạt, ngược lại, nếu tài khoản phù hợp với loại nhân viên là Quản lý, trang quản lý sẽ được kích hoạt). Ngược lại, nếu tài khoản không hợp lệ, trang báo lỗi sẽ hiện ra, yêu cầu nhân viên nhập lại tên tài khoản và mật khẩu.

Theo đó, mỗi trang nhân viên tương ứng sẽ thể hiện các chức năng tương ứng mà họ được phép thực hiện<sup>40</sup>

Khi không sử dụng chương trình nữa, ứng dụng sẽ tự động đăng xuất.

### 6.2.2. Danh sách các màn hình xử lý chính

STT	Tên màn hình	Ý nghĩa/Ghi chú
1	Đăng Nhập	Cho phép nhân viên đăng nhập với tài khoản hợp lệ trước khi sử dụng chương trình
2	Trang Người Dùng	Màn hình làm việc chính của nhân viên
3	Trang Quản Lý	Màn hình làm việc chính của Quản lý

<sup>40</sup> Xem phụ lục: Mô tả chi tiết các màn hình xử lý trong ứng dụng với một số dữ liệu mẫu

4	TrangBáoLỗi	Trang thông báo các lỗi xảy ra trong quá trình sử dụng ứng dụng
---	-------------	---

**Bảng 47. Danh sách các màn hình xử lý chính**

## 7. Giải pháp kĩ thuật

Trong quá trình thực hiện chương trình, đa số các chức năng đều được thiết kế sử dụng hai nền tảng trong bộ .Net Framework 3.5 của Microsoft. Tuy nhiên, hai chức năng quan trọng nhất trong ứng dụng là hiển thị LCV dạng sơ đồ và thống kê tình trạng các đợt thi đang tiến hành dưới dạng sơ đồ Gantt có thể xem là hai chức năng khá phức tạp. Để thiết kế được hai chức năng này, chúng tôi đã sử dụng một số kĩ thuật do chúng tôi tự đề xuất. Đồng thời, vì để đáp ứng đúng đặc điểm của LCV tổ chức thi, chúng tôi đã thiết kế riêng đối tượng Custom Activity, gọi là WorkItem để làm cơ sở cho các đơn vị công việc con trong LCV.

### 7.1. Chức năng hiển thị luồng công việc dạng sơ đồ:

Để xây dựng chức năng này, chúng tôi đã viết thêm thư viện hỗ trợ cho việc vẽ sơ đồ luồng công việc lên giao diện.

- Thư viện ModelReader: dùng để đọc file XOML và các thông tin của các đối tượng trong file XOML (bao gồm các đối tượng WorkItem, đối tượng Parallel, đối tượng Sequence). Thư viện này còn cho chúng tôi biết được cấu trúc file XOML bao gồm các cấu trúc cha và con của các đối tượng bất kỳ.
- Thư viện ModelLayouter: dùng để quy định tọa độ và chiều dài, chiều rộng của các đối tượng trong file XOML. Sau khi có được cấu trúc của file XOML thì thư viện này sẽ tính toán tọa độ hiển thị của các đối tượng lên trên giao diện.

Ngoài ra, chúng tôi thiết kế thêm ba UserControl tương ứng với ba đối tượng WorkItem, Parallel, Sequence để có thể vẽ sơ đồ LCV lên trên giao diện. Ba

UserControl này sẽ lấy tọa độ từ thư viện ModelLayouter để vẽ lên trên giao diện các đối tượng tương ứng với các thuộc tính và tọa độ được quy định sẵn trong file XOML. Nếu mỗi đối tượng WorkItem, Parallel hay Sequence có các đối tượng con thì các UserControl đó sẽ tự gọi các UserControl tương ứng với các đối tượng con để vẽ các chúng lên trên giao diện. Từ đó chúng tôi đã có thể hiển thị được sơ đồ luồng công việc lên trên giao diện một cách tùy biến.

#### Đặc điểm:

- Sử dụng định nghĩa LCV bằng XAML có sẵn của WF làm ngôn ngữ MHH LCV
- Chỉ thể hiện các đối tượng chính mà người dùng quan tâm
- Độc lập với nghiệp vụ. Chức năng này hoàn toàn linh động, có thể đọc được bất cứ LCV nào được thiết kế bằng ngôn ngữ XAML và sử dụng tối tượng cơ bản là WorkItem do chúng tôi thiết kế
- Tương tác kéo thả: Hỗ trợ người dùng có thể kéo thả vào sơ đồ. Đặc điểm này được thể hiện trong chức năng phân công người thực hiện vào LCV
- Hỗ trợ phóng to, thu nhỏ sơ đồ: Cho phép người dùng điều chỉnh kích thước của sơ đồ sao cho phù hợp nhất với chính mình
- Linh động, hỗ trợ nhiều nhu cầu khác nhau: Phân công, cập nhật kết quả thực hiện tác vụ đối với người nhân viên thửa hành...

#### 7.2. Chức năng thống kê tình trạng các đợt thi đang tiến hành:

Để xây dựng chức năng này, chúng tôi tạo ra “Sơ đồ Gantt” trên nền tảng Grid (một control được WPF xây dựng sẵn). Về cơ bản, chúng tôi đã xây dựng các hàm nhằm lấy lên thời điểm bắt đầu và thời điểm kết thúc của tất cả các đợt thi đang diễn ra, từ đó tính toán số lượng cột để vẽ lên trên Grid. Sau đó, dựa trên thời điểm bắt

đầu và thời gian thực hiện của mỗi đợt thi để có thể hiển thị được thời gian đợt thi này đang diễn ra và đã diễn ra được trong bao lâu. Đồng thời, ứng với mỗi đợt thi sẽ có ngày bắt đầu và ngày kết thúc nên chúng tôi sẽ có được các chỉ số tương ứng so với ngày bắt đầu của thời gian trong “Sơ đồ Gantt” để có thể hiện thị lên thời gian diễn ra và thông tin chi tiết của các đợt thi tương ứng. Từ đây chúng tôi có thể xây dựng “Sơ đồ Gantt” với giao diện và hiển thị thông tin một cách tùy biến theo nhu cầu sử dụng.

Đặc điểm:

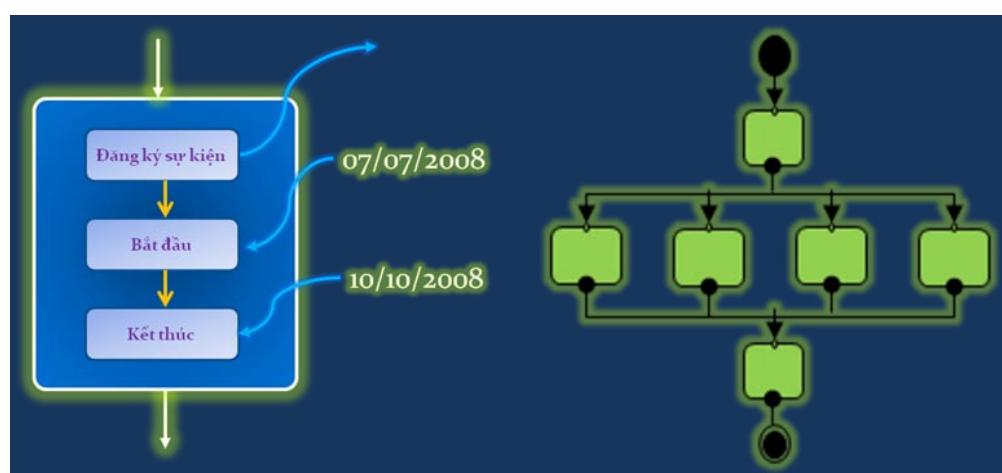
- Thiết kế dựa trên control GridView của WPF
- Đọc tự động đối tượng công việc và thời gian lên sơ đồ
- Hỗ trợ chức năng cảnh báo bằng các màu khác nhau tương ứng với các tình trạng thực thi của từng thẻ hiện LCV, cũng như các công việc tương ứng trong các thẻ hiện đó.
- Linh động, tùy biến

### 7.3. Custom Activity

Như đã trình bày ở phần trước, WF hỗ trợ nhiều dịch vụ khác nhau để xây dựng, quản lý và thực thi LCV. Với *Persistency Service*, ứng dụng quản lý LCV có thể bảo đảm cho một LCV bất kỳ thực hiện trong một khoảng thời gian dài. Sau mỗi phiên làm việc, ứng dụng có thể được tắt và bật lại trong các phiên làm việc sau, điều đó không ảnh hưởng tới việc thực thi một LCV. Tuy nhiên, không có dịch vụ nào hỗ trợ việc thực thi một công việc thành phần của LCV trong một khoảng thời gian cho trước. Cụ thể hơn, bản thân WF chỉ cho phép thực hiện một số xử lý trong mỗi công việc, và bắt cứ khi nào các xử lý này kết thúc thì công việc cũng kết thúc, các công việc tiếp sau đó sẽ được tự động bắt đầu.

Với các LCV tổ chức thi, các ràng buộc về thời gian thực hiện công việc là cực kỳ quan trọng. Từng công việc trong LCV này không thể chỉ diễn ra trong vòng vài giây. Từng công việc như tổ chức thi, chấm thi, cấp chứng chỉ v.v... có các mốc thời gian cụ thể để thực hiện. Thậm chí, đơn vị thời gian dùng trong công tác quản lý của lĩnh vực này là đơn vị ngày hoặc buổi chứ không chi tiết đến từng giây. Giờ làm việc của các nhân viên trực tiếp thực hiện công việc là giờ hành chính và bản thân họ cũng chỉ cần quan tâm công việc do họ phụ trách có ngày bắt đầu và ngày kết thúc là ngày nào mà không cần quan tâm đến mức giờ, phút, giây... Ngoài ra, như đã giới thiệu, LCV tổ chức thi có những công việc đặc biệt mà thời gian thực hiện công việc đó phụ thuộc hoàn toàn vào các hệ thống bên ngoài. Chính bản thân người thực hiện công việc không chủ động quản lý được. Chính vì thế, việc quản lý thực thi công việc bằng phương pháp đơn giản như hẹn giờ hay tính tổng thời gian thực hiện không có hiệu quả.

Giải pháp được sử dụng là tự thiết kế lại một đối tượng công việc, dựa trên các Activity cơ bản đã hỗ trợ sẵn. Mỗi công việc sẽ có 2 activity bên trong chờ nhận sự kiện bắt đầu hoặc kết thúc từ người thực hiện công việc. Khi đó, việc quản lý thực thi công việc sẽ trở nên linh động hơn, dựa vào hành động thực tế từ người sử dụng của ứng dụng.



**Hình 25. WorkItem**

Thêm một vấn đề nữa, tất cả các công việc trong LCV tổ chức thi có cấu trúc như nhau. Nghĩa là đều nhận 2 sự kiện bắt đầu và kết thúc công việc như nhau. Chính vì lý do đó, khi có nhiều công việc cùng chờ nhận cùng một sự kiện giống nhau, vậy khi sự kiện đó được chuyển tới LCV, công việc nào sẽ nhận được sự kiện ấy? Bản thân WF không hỗ trợ việc này, khi có nhiều công việc cùng chờ nhận cùng một sự kiện, thì công việc đầu tiên trong nhóm các công việc song song đó sẽ luôn nhận được sự kiện. WF mặc định chuyển sự kiện cho công việc đầu tiên trong hàng đợi. Giải pháp cho vấn đề này là mỗi công việc trước khi nhận sự kiện sẽ thực hiện gửi một thông điệp ra ngoài, tạm gọi là đăng ký nhận sự kiện thông qua một mã duy nhất. Dựa vào mã này, ứng dụng chủ có thể gửi sự kiện đến đích danh công việc mà người dùng đang tương tác.

## **Chương 5: Tổng kết**

*Trong chương này, chúng tôi sẽ tổng kết các kết quả đã đạt được sau quá trình nghiên cứu lý thuyết và xây dựng ứng dụng thực tế. Đồng thời, chương cũng sẽ nêu ra một số ưu, khuyết điểm của ứng dụng, những khó khăn mà chúng tôi đã gặp phải trong suốt quá trình xây dựng, cũng như hướng phát triển trong tương lai đối với ứng dụng mà chúng tôi đã xây dựng.*

### **1. Tóm tắt công việc đã làm**

Ngay từ đầu, sau khi nghiên cứu về các chuẩn mô MHH LCV được đưa ra bởi WfMC cũng như nền tảng WF mà Microsoft đã xây dựng hỗ trợ trong việc xây dựng ứng dụng, chúng tôi đã xác định sẽ xây dựng phần mềm trên nền .Net với WF. Trên thực tế, có nhiều Engine hỗ trợ thiết kế ứng dụng LCV dưới nhiều hình thức khác nhau được xây dựng trên nền WF như Sharepoint Workflow, Silverlight, ASP.Net... Tuy nhiên, do đặc điểm của ứng dụng và yêu cầu của đề cương là phải thử nghiệm trên một nền công cụ khác với nền công cụ đã sử dụng trong những khóa luận trước đây, chúng tôi đã quyết định xây dựng một ứng dụng chạy trên mạng cục bộ chứ không xây dựng ứng dụng Web trên trình duyệt, đồng thời sử dụng công nghệ WPF cũng do Microsoft cung cấp trên bộ .Net 3.5 làm công nghệ thiết kế giao diện và xây dựng chương trình tích hợp với WF. Vì vậy, chúng tôi đã quyết định sử dụng ngôn ngữ C# làm ngôn ngữ lập trình chính cho ứng dụng. Chương trình có thể thực thi trực tiếp trên máy tính có hệ điều hành Windows được cài sẵn .Net Framework 3.5 mà không cần đến quá trình cài đặt phức tạp nào.

## **2. Đặc điểm của Công việc**

### **2.1. Nội dung**

Trong suốt quá trình thực hiện khóa luận, thông qua tìm hiểu thực tế nghiệp vụ quản lý quy trình, cụ thể là quy trình thi, cấp chứng chỉ và in văn bằng cho các đơn vị đào tạo chứng chỉ quốc gia, chúng tôi đã thiết kế một ứng dụng mà theo chúng tôi là đã có khả năng đáp ứng các nhu cầu hiện tại của nghiệp vụ. Ứng dụng được hoàn thành với các ưu điểm sau:

- Ứng dụng có khả năng thể hiện tốt quy trình dưới dạng sơ đồ LCV rất trực quan và rất dễ hiểu. Hơn nữa, ở đây, việc vẽ nên LCV không phải được thiết kế cứng nhắc với quy trình thi, cấp chứng chỉ và in văn bằng; mà với bất kỳ quy trình nào được thiết kế thừa các định dạng của chúng tôi thì chương trình đều có thể thể hiện được.
- Ứng dụng cũng thể hiện tốt lược đồ các thể hiện của quy trình dưới dạng sơ đồ Gantt vốn gắn bó với người Quản lý. Vì vậy, người sử dụng chương trình không cần phải tìm hiểu nghiên cứu cách sử dụng một cách quá khắt.
- Với các chức năng thống kê, ứng dụng đưa ra các tiêu chí rất tổng quát, đồng thời cho phép thêm các tiêu chí mới phù hợp với các nhu cầu của người quản lý.
- Tạo nền tảng cho việc có thể phát triển WF View , Gantt Viewvà SQL Generator.
- Ứng dụng còn kết hợp thêm các chức năng kiểm soát hoạt động của các nhân viên như: gửi thông báo nhắc nhở tự động khi nhân viên chưa kích hoạt các tác vụ trên ứng dụng; buộc nhân viên phải nộp bảng báo cáo kết quả thực hiện công việc trước khi báo cáo kết thúc công việc đang thực hiện; kiểm tra các thời gian thực hiện từng công việc nhằm đánh giá tình trạng của công

việc (sớm/ trễ hay đúng thời hạn); cho phép nhân viên cập nhật nguyên nhân khi không hoàn thành công việc đúng thời gian.

- Tích hợp chức năng bảo mật tài khoản người dùng thông qua các phương thức bảo mật.

Tuy nhiên, thông qua tìm hiểu một cách chi tiết độ phức tạp của quy trình, chúng tôi hiểu ứng dụng mà chúng tôi đã xây dựng trong khóa luận này thật sự còn tồn tại nhiều hạn chế:

- Ứng dụng được thiết kế với mỗi đơn vị LCV nhỏ nhất là đợt thi. Tuy nhiên trên thực tế, mỗi đợt thi có thể được tổ chức tại nhiều cơ sở khác nhau, với số lượng thí sinh khác nhau. Vì vậy thời gian thực hiện các công việc tương ứng, cũng như các phân công nhân sự... cũng khác nhau (chẳng hạn như đối với một đợt thi tổ chức tại hai tỉnh thành khác nhau, thì tỉnh thành ở xa cần được phân công thực hiện trước để có thể gửi kết quả về kịp cho thí sinh). Do đó, theo thực tế quy trình, các đơn vị LCV nhỏ nhất nên là địa điểm thi chứ không phải là đợt thi. Tuy nhiên do thời gian hạn chế và khả năng hỗ trợ của WF mà chúng tôi chỉ thiết kế đến đơn vị đợt thi mà không chia nhỏ hơn nữa.
- Ứng dụng thiết kế đối với một LCV cụ thể. Tuy các chức năng thể hiện sơ đồ được thiết kế động, nhưng toàn chương trình tạm thời chỉ có thể đáp ứng các quy trình thi trong nghiệp vụ quản lý đào tạo (do cơ sở dữ liệu chỉ phù hợp với nghiệp vụ này). LCV được định nghĩa sử dụng định dạng XAML do .Net Framework của Microsoft hỗ trợ, được thiết kế sử dụng công cụ Workflow Designer tích hợp trong Visual Studio. Vì thế, tập tin định nghĩa tuy có dạng cấu trúc, nhưng vẫn rất khó tiếp cận đối với người dùng. Khi cần thay đổi LCV khác, đòi hỏi ứng dụng phải được nâng cấp ở mức thiết kế.

- Chưa cho phép người dùng tùy biến LCV theo nhu cầu như thêm, bớt công việc trong LCV, mà chỉ hỗ trợ thay đổi các thuộc tính của bản thân công việc như: tên công việc, thời gian thực hiện.
- Giao diện và chức năng chưa hoàn toàn hướng người dùng.
- Chưa giải quyết được hiện tượng thắt cổ chai khi có quá nhiều thẻ hiện LCV cùng diễn ra một lúc, dẫn đến khối lượng công việc quá nhiều gây khó khăn cho người quản lý do thời gian thực hiện khóa luận có giới hạn.

Vì vậy, chúng tôi rất mong chương trình này có thể tiếp tục được cải thiện trong tương lai, để có thể hỗ trợ tốt hơn cho người quản lý trong việc thực hiện nghiệp vụ của mình.

## **2.2. Hướng phát triển nhằm giải quyết những hạn chế về mặt nội dung**

Như đã đề cập ở phần 2.1, ứng dụng được thiết kế trong luận văn này còn tồn tại nhiều hạn chế. Tuy nhiên, những hạn chế này không phải không giải quyết được. Vì vậy, trong phần này, chúng tôi sẽ đề cập đến hướng phát triển tương lai cho ứng dụng, nhằm giải quyết các hạn chế kể trên và hỗ trợ tốt hơn cho người dùng:

- Ứng dụng có thể xây dựng một cách tổng hợp. Nghĩa là không chỉ đối với quy trình thi, mà bất cứ quy trình nghiệp vụ nào cũng có thể sử dụng ứng dụng này. Từ đó, ứng dụng có khả năng cho phép người dùng thiết lập ngay trên giao diện quy trình của mình, các chức năng họ mong muốn trong chương trình; cho phép thêm bớt các chức năng khác nhau theo nhu cầu sử dụng; hoặc thay đổi các thông số mặc định của ứng dụng một cách trực quan
- Giao diện hoàn toàn kéo thả, cập nhật kết quả tự động.
- Kết quả cập nhật liên tục; nghĩa là khi một công việc được hoàn thành sẽ cập nhật ngay trên giao diện quản lý, có thể dưới dạng thông báo, email, hoặc dạng thông tin trực tiếp trên sơ đồ Gantt

- Có thể xây dựng thêm ứng dụng web và ứng dụng di động, giúp người quản lý có thể theo dõi tiến độ công việc, và người nhân viên có thể cập nhật thông tin kết quả công việc của mình mọi lúc mọi nơi.
- Có thể sử dụng WF4, được tích hợp trong bộ .Net Framework 4 được Microsoft phát hành tháng 4/2010, sử dụng Visual Studio 10 để xây dựng Ứng dụng mới (nâng cấp từ chương trình hiện tại). Vì theo chúng tôi tìm hiểu, tuy vẫn có khả năng tương thích với các WF cũ (3.0 và 3.5) - nghĩa là không cần thiết kế lại ứng dụng từ đầu, nhưng WF4 được thiết kế với bộ Engine hoàn toàn mới, đổi mới các dịch vụ hỗ trợ (Hosting, Persistence...), các đơn vị Activity, lưu trữ dữ liệu.... cho phép xử lý các vấn đề còn hạn chế ở WF3.0 và WF3.5.
- Mở rộng, cho phép ứng dụng có thể cùng lúc quản lý nhiều LCV khác nhau, chứ không chỉ gói gọn trong LCV tổ chức thi

Như vậy, với các hướng phát triển đã đề nghị, chúng tôi hi vọng ứng dụng có thể phát triển tốt hơn trong tương lai, để người quản lý Việt Nam và các doanh nghiệp nước nhà có thể sử dụng chương trình tốt nhất hoàn toàn miễn phí.

## LỜI KẾT

Cuối cùng, sau gần 6 tháng thực hiện, khóa luận cũng đã đến hồi kết. Khoảng thời gian thực hiện khóa luận tốt nghiệp này không quá dài nhưng lại nảy sinh rất nhiều khó khăn; từ việc lựa chọn hướng đi, tìm hiểu công nghệ với những lần đi vào ngõ cụt, hay những khó khăn vấp phải khi làm việc với bạn đồng hành, khó khăn trong cuộc sống, công việc riêng,...nhưng cuối cùng, trên tất cả chúng tôi đã vượt qua và đi đến đích cuối cùng. Chúng tôi rất cảm ơn PGS.TS. Đồng Thị Bích Thủy, Giáo viên hướng dẫn của chúng tôi, đã cùng chúng tôi vượt qua chặng đường khó khăn này. Nhờ có những lời động viên khích lệ, sự hướng dẫn nhiệt thành của cô chúng tôi mới tìm được tiếng nói chung và vạch ra được hướng đi cho đến ngày hôm nay.

Ngoài ra, chúng tôi cũng xin gửi lời cảm ơn đến Trung Tâm Tin Học, trường Đại Học Khoa Học Tự Nhiên đã tạo mọi điều kiện để chúng tìm hiểu bối cảnh thực tế của quy trình tổ chức thi và cấp chứng chỉ quốc gia tại trung tâm.

Và chúng tôi cũng xin gửi lời cảm ơn đến nhau, bạn đồng hành thân mến! Trải qua những khó khăn, khúc mắc chúng ta càng lúc càng trưởng thành hơn, chín chắn hơn.

Cuối cùng, chúng tôi hi vọng khóa luận này không chỉ dừng lại ở đây, một lúc nào đó, những điều mà chúng tôi trăn trở, chưa làm được sẽ được mang ra đào sâu, nghiên cứu tiếp.

Xin chân thành cảm ơn!

Nguyễn Minh Bình – Diệp Huỳnh Anh

## **Tài liệu tham khảo**

1. Kenn Scribner, Microsoft Windows Workflow Foundation Step by Step. A Division of Microsoft Corporation, One Microsoft Way Redmond, Washington 98052-6399, 2007
2. Lê Nhựt Minh, Nguyễn Trần Minh Tú, Tìm hiểu và ứng dụng Windows Workflow Foundation để hỗ trợ các quy trình nghiệp vụ, Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, TpHCM, 2009
3. SHI Meilin, YANG Guangxin, XIANG Yong, WU Shangguang, Workflow Management Systems: A Survey, Department of Computer Science, Tsinghua University, Beijing, P.R.China 100084, 1998.
4. Võ Hữu Phúc, Đào Anh Vũ, Nghiên cứu giải pháp nguồn mở cho Workflow quản lý hồ sơ công văn, Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, TpHCM, 2008

## PHỤ LỤC

*Trong phần này, chúng tôi sẽ trình bày một số chi tiết phụ mà chúng tôi nghĩ sẽ giúp ích cho người đọc với nhu cầu tìm hiểu thêm cả về mặt lý thuyết cũng như thực tế ứng dụng. Phần này sẽ bao gồm các nội dung sau:*

### 1. XML Schema

```
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wf="http://www.wfmc.org/wFXML/2.0/wFXML20.xsd"
  xmlns:as="http://www.oasis-open.org/asap/0.9/asap.xsd"
  targetNamespace="http://www.wfmc.org/wFXML/2.0/wFXML20.xsd"
  elementFormDefault="qualified">
  <xsd:import namespace="http://www.oasis-open.org/asap/0.9/asap.xsd"
    schemaLocation="http://docs.oasis-open.org/asap/0.9/asap.xsd" />
  <xsd:annotation>
    <xsd:documentation
      xml:lang="en">=====
      = WfXML.XSD
      ===== DRAFT
      Original 2005.02.15, Chi-Tsai Yang Revised 2005.02.25 Sameer Pradhan
    </xsd:documentation>
  </xsd:annotation>
```

#### Removed

```
  redundant definitions</xsd:documentation>
</xsd:annotation>
  <xsd:simpleType name="PortTypes">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Instance" />
      <xsd:enumeration value="Factory" />
```

```

<xsd:enumeration value="Observer" />
<xsd:enumeration value="Activity" />
<xsd:enumeration value="ServiceRegistry" />
</xsd:restriction>
</xsd:simpleType>

<xsd:element name="processDefinitionType">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ProcessLanguage" type="xsd:string" />
<xsd:element name="Definition">
<xsd:complexType>
<xsd:sequence>
<xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
- <!--
properties =====
-->
<xsd:group name="serviceRegistryPropertiesGroup">
<xsd:sequence>
<xsd:element name="Key" type="xsd:anyURI" />
<xsd:element name="Name" type="xsd:string" />
<xsd:element name="Description" type="xsd:string" />
<xsd:element name="Version" type="xsd:string" />

```

```

<xsd:element name="Status" type="xsd:string" />
</xsd:sequence>
</xsd:group>
<xsd:group name="activityPropertiesGroup">
<xsd:sequence>
<xsd:element name="Key" type="xsd:anyURI" />
<xsd:element name="State" type="as:stateType" />
<xsd:element name="Name" type="xsd:string" />
<xsd:element name="Description" type="xsd:string" />
<xsd:element name="ValidStates">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ValidState" type="as:stateType" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="InstanceKey" type="xsd:anyURI" />
<xsd:element name="RemoteInstance" type="xsd:anyURI" />
<xsd:element name="StartedDate" type="xsd:dateTime" />
<xsd:element name="DueDate" type="xsd:dateTime" />
<xsd:element name="LastModified" type="xsd:dateTime" />
</xsd:sequence>
</xsd:group>
- <!--
messages =====
-->
<xsd:element name="GetDefinitionRq">
<xsd:complexType>

```

```

<xsd:sequence>
<xsd:element name="ProcessLanguage" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="GetDefinitionRs">
<xsd:complexType>
<xsd:sequence>
<xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
- <!--
(2) WfXML20 p.14
-->
<xsd:element name="NewDefinitionRq">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="wf:processDefinitionType" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="NewDefinitionRs">
<xsd:complexType>
<xsd:sequence>
<xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>

```

```

</xsd:complexType>
</xsd:element>
<xsd:element name="SetDefinitionRq">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="wf:processDefinitionType" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="SetDefinitionRs">
<xsd:complexType>
<xsd:sequence>
<xsd:any namespace="#any" processContents="lax" minOccurs="0"
maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ListDefinitionsRq" />
<xsd:element name="ListDefinitionsRs">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="wf:DefinitionInfo" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="DefinitionInfo">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="DefinitionKey" type="xsd:string" />

```

```

<xsd:element name="Name" type="xsd:string" minOccurs="0" />
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
<xsd:element name="Version" type="xsd:string" minOccurs="0" />
<xsd:element name="Status" type="xsd:string" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ListActivitiesRq" />
  <xsd:element name="ListActivitiesRs">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="wf:ActivityInfo" minOccurs="0" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ActivityInfo">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ActivityKey" type="xsd:anyURI" />
        <xsd:element name="Name" type="xsd:string" minOccurs="0" />
        <xsd:element name="Description" type="xsd:string" minOccurs="0" />
        <xsd:element name="Assignee" type="xsd:string" minOccurs="0"
maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="CompleteActivityRq">
    <xsd:complexType>
      <xsd:sequence>

```

```
<xsd:element name="Option" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="CompleteActivityRs" />
</xsd:schema>
```

## 2. Định dạng mẫu tập tin Designer trong WF

```
using System;
using System.ComponentModel;
using System.ComponentModel.Design;
using System.Collections;
using System.Drawing;
using System.Reflection;
using System.Workflow.ComponentModel.Compiler;
using System.Workflow.ComponentModel.Serialization;
using System.Workflow.ComponentModel;
using System.Workflow.ComponentModel.Design;
using System.Workflow.Runtime;
using System.Workflow.Activities;
using System.Workflow.Activities.Rules;

namespace PCodeFlow
{
    partial class Workflow1
    {
        #region Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// 
    }
}
```

```

/// the contents of this method with the code editor.

/// </summary>

[System.Diagnostics.DebuggerNonUserCode]
private void InitializeComponent()
{
    this.CanModifyActivities = true;
    System.Workflow.Activities.CodeCondition codecondition1 = new
System.Workflow.Activities.CodeCondition();
    System.Workflow.Activities.CodeCondition codecondition2 = new
System.Workflow.Activities.CodeCondition();
    this.codeActivity2 = new System.Workflow.Activities.CodeActivity();
    this.codeActivity1 = new System.Workflow.Activities.CodeActivity();
    this.ifElseBranchActivity2 = new
System.Workflow.Activities.IfElseBranchActivity();
    this.ifElseBranchActivity1 = new
System.Workflow.Activities.IfElseBranchActivity();
    this.ifElseActivity1 = new System.Workflow.Activities.IfElseActivity();
    //
    // codeActivity2
    //
    this.codeActivity2.Name = "codeActivity2";
    this.codeActivity2.ExecuteCode += new
System.EventHandler(this.PostalCodeInvalid);
    //
    // codeActivity1
    //
    this.codeActivity1.Name = "codeActivity1";
    this.codeActivity1.ExecuteCode += new
System.EventHandler(this.PostalCodeValid);

```

```

//  

// ifElseBranchActivity2  

//  

this.ifElseBranchActivity2.Activities.Add(this.codeActivity2);  

codecondition1.Condition += new  

System.EventHandler<System.Workflow.Activities.ConditionalEventArgs>(this.Ev  

aluatePostalCode);  

this.ifElseBranchActivity2.Condition = codecondition1;  

this.ifElseBranchActivity2.Name = "ifElseBranchActivity2";  

//  

// ifElseBranchActivity1  

//  

this.ifElseBranchActivity1.Activities.Add(this.codeActivity1);  

codecondition2.Condition += new  

System.EventHandler<System.Workflow.Activities.ConditionalEventArgs>(this.Ev  

aluatePostalCode);  

this.ifElseBranchActivity1.Condition = codecondition2;  

this.ifElseBranchActivity1.Name = "ifElseBranchActivity1";  

//  

// ifElseActivity1  

//      this.ifElseActivity1.Activities.Add(this.ifElseBranchActivity1);  

this.ifElseActivity1.Activities.Add(this.ifElseBranchActivity2);  

this.ifElseActivity1.Name = "ifElseActivity1";  

//  

// Workflow1  

//  

this.Activities.Add(this.ifElseActivity1);  

this.Name = "Workflow1";  

this.CanModifyActivities = false;

```

```

    }

#endregion

private IfElseBranchActivity ifElseBranchActivity2;
private IfElseBranchActivity ifElseBranchActivity1;
private CodeActivity codeActivity1;
private CodeActivity codeActivity2;
private IfElseActivity ifElseActivity1;

}

}

```

### 3. Các sản phẩm ứng dụng sử dụng XPDL<sup>41</sup>

A	<b>Active Endpoint's ActiveVOS</b> visual orchestration system supports XPDL 2.1  <b>ActiveModeler</b> , see "KAISHA-Tec"  Adobe has sucessfully implemented XPDL within <u>Adobe LiveCycle Workflow</u>  <b>ADVANTYS WorkflowGen</b> supports XPDL  <b>Amazonas Workflow</b> is a Java based workflow engine which supports XPDL  <b>Arachnea EverSuite</b> supports XPDL  <u>Appian Enterprise</u> and <u>Zynium's Byzio</u> use XPDL for interchange of Visio process models  <b>Ascentn AgilePoint Server</b> is a .NET-based BPMS that supports XPDL  <b>Aspose's Aspose.Workflow</b> is a .Net workflow engine using XPDL  <b>Assetlink Corporation</b> uses XPDL to define and store processes in <b>Marketing Workbench</b>
B	<u>BOC ADONIS 3.7</u> (and higher) supports XPDL export

<sup>41</sup> <http://www.wfmc.org/xpdl-implementations.html>

	<p>BEA Systems supports XPDL in the <a href="#">AquaLogic Enterprise Repository</a> and <a href="#">BPM Suite</a></p>
	<p><b>Brein VB's InProces</b> uses XPDL 2.0</p>
	<p><b>Bonita</b> is an open source workflow solution using XPDL</p>
	<p><b>ProEd Workflow Editor</b> is a XPDL compliant design tool on top of Bonita</p>
C	<p><b>Canto CanFlow</b> uses XPDL within this Digital Asset Management solution</p>
	<p><b>CapeVisions</b> supports XPDL including a free plugin to Visio that edits XPDL</p>
	<p><b>CHALEX BPM Framework</b> supports XPDL</p>
	<p><b>ComActivity</b> supports XPDL in its process design tool and runtime engine</p>
	<p><u>Cordys BPMS</u> supports XPDL for process definition import and export</p>
	<p><b>COSA Designer and the COSA BPM engine</b> support XPDL</p>
	<p>Cubetto Toolset is a generic modelling tool which can export XPDL</p>
D	<p>Documentum, see "EMC"</p>
E	<p><b>Eclaire Group Lynx Flow Designer</b> supports XPDL</p>
	<p><b>EMC Documentum ApplicationXtender Workflow</b> supports import and export of XPDL</p>
	<p><b>EMC Documentum Process Suite</b> supports XPDL</p>
	<p><b>Enhydra Shark</b> is an open source XPDL workflow engine in Java</p>
	<p><b>Enhydra JaWE</b> an openSource graphical XPDL workflow editor</p>
	<p><b>First Trace's Kinnosa Workflow</b> supports XPDL for process model exchange</p>
F	<p><b>Finantix Studio FXS</b> supports BPMN and XPDL for Business Process Modelling</p>

	<p><u>Fujitsu Interstage BPM (i-Flow)</u> supports XPDL and BPMN</p>
	<p><b>FileNet Business Process Manager 4.0</b> supports XPDL 1.0 and 2.0 as well as BPMN</p>
G	<p><u>Global 360 Business Optimization Server (BOS)</u> supports XPDL 2.0 and BPMN</p>
	<p>GlobalSight, see "Transware"</p>
H	<p><b>HOGA.PL'S intraDok</b> supports import and export of process definitions in XPDL</p>
I	<p><b>IBM FileNet Business Process Manager 4.0</b> supports XPDL 1.0 and 2.0 as well as BPMN</p> <p><b>IDS Scheer Business Architect</b> supports export of process models to XPDL through an optional add-on</p> <p><b>iGrafx</b> supports XPDL 2.1 within iGrafx 2009 (import and export)</p> <p><b>Interwoven WorkRoute MP</b> supports XPDL for both import and export.</p> <p><b>Infinity Process Engine</b> supports XPDL for import/export and Wf-XML 2.0 deployment</p> <p><b>Infor (formerly SSA Global)</b> supports XPDL in a BPM engine within its ERP suite</p> <p><b>ITP-Commerce Design</b> provides a XPDL 1.0 validation module</p>
J	<p><b>jawFlow</b> is an open source workflow engine supports XPDL</p> <p><b>Jenz &amp; Partner's BPEdit</b> is an ontology-based business process editor that supports XPDL</p> <p>KAISHA-Tec's <b>ActiveModeler Avantage</b> supports XPDL 2.0</p>
L	<p><u>Lombardi's Blueprint</u> supports XPDL 2.1 and BPMN</p>
M	<p><b>Metoda S.p.A OpenMet BPMF</b> supports XPDL</p> <p><b>Mono-sys's Tigris BPM</b> solution supports XPDL 2.0 as well as BPMN</p>
N	<p><b>Nautica</b> uses XPDL process definition data with GUI based definition editor</p>

O	<p><b>Open Business Engine</b> is an open source Java workflow engine based on XPDL</p> <p><b>OpenPages Governance Platform</b> solution for enterprise-wide business governance using XPDL</p> <p><u>Openwork</u> is validating support for XPDL and Wf-XML</p> <p><b>Oracle 9i Warehouse Builder 9.2</b> saves process definitions in XPDL</p>
P	<p><b>Pentaho's B1 Platform</b> uses an XPDL-based workflow engine to execute activities within the system</p> <p><u>Projekty Bankowe Polsoft's BPB Workflow</u> supports import/export in XPDL 2.0</p>
Q	<p><b>QualiWare</b> supports XPDL 2.1 for both import and export of process models</p>
R	<p><b>R-Data's E-SOD</b> business process export using XPDL 2.0 and import using version 1.0 and above</p> <p><b>Rodan Systems OfficeObjects Workflow</b> is an embedded, commercial workflow engine using XPDL</p>
S	<p><u>Savvion</u> supports XPDL for import and export of process models through its Process Modeler</p> <p><u>Simprocess</u> from CACI supports XPDL for simulation models</p> <p><b>Software AG's Crossvision BPM</b> supports XPDL 1.0 and XPDL 2.0</p> <p><b>SpeechCycle's LevelOne</b> virtual CSR platform uses XPDL</p> <p>SSA Global, see "Infor"</p>
T	<p>Tell-Eureka, see "SpeechCycle"</p> <p><u>TIBCO iProcess Suite</u> supports XPDL</p> <p><b>Together Workflow Editor</b> is a graphical XPDL-based workflow editor</p> <p><b>Transware Ambassador</b> embeds an XPDL-compliant workflow engine</p>
U	<p><u>Unisys</u> has done significant BPM development using XPDL</p>
V	<p><b>Vignette Process Workflow Modeler</b> supports XPDL</p>

W	<p>a. <u>W4's W4 BPM Suite supports XPDL</u></p> <p><b>WfMOpen</b> is an open source workflow engine that uses XPDL</p> <p><u>Workflow::Wfmc</u> is an OpenSource lightweight Workflow Engine in PERL based on XPDL 2.0</p>
Z	<p><u>Zynium's Byzio</u> is a Visio plugin enabling two-way transformation of Visio diagrams and XPDL</p>

#### 4. Mô tả chi tiết từng màn hình xử lý chính

##### 4.1. Màn hình Đăng nhập:



Hình 26. Màn hình đăng nhập

Mô tả các Control trên màn hình:

STT	Nội dung	Mô tả
1	Tab Đăng nhập	Thẻ chứa nội dung đăng nhập
2	Tên người dùng	Tên người dùng
3	Mật khẩu	Mật khẩu tương ứng giúp người dùng đăng nhập ứng dụng
4	Button Đăng Nhập	Kiểm tra giá trị tên người dùng và mật khẩu chính xác không. Nếu chính xác, ứng dụng sẽ chuyển đến trang giao diện tương ứng với người sử dụng ứng dụng, nếu không, sẽ báo lỗi
5	Tab Cơ sở dữ liệu	Thẻ chứa nội dung cài đặt cho cơ sở dữ liệu đang

		sử dụng, cho phép người dùng thay đổi giá trị thích hợp để có thể đăng nhập đúng vào cơ sở dữ liệu của ứng dụng
6	Database	Combobox chứa các cơ sở dữ liệu khác nhau mà ứng dụng sử dụng trong quá trình chạy
7	Tên server	Server chứa cơ sở dữ liệu
8	Tên database	Tên database tương ứng trên server
9	Tên truy cập	Tên dùng đăng nhập database
10	Mật khẩu truy cập	Mật khẩu đăng nhập database
11	Button Lưu	Thay đổi các giá trị mặc định trên tập tin chứa thông tin server
12	Button Hủy	Không lưu, reset lại giá trị trước đó
13	Button Reset	Reset lại các giá trị từ tập tin đã lưu

**Bảng 48. Danh sách control màn hình đăng nhập**

#### 4.2. Trang Báo lỗi:



**Hình 27. Trang báo lỗi**

Mô tả: Đây là màn hình thông báo các lỗi xảy ra trong quá trình làm việc của nhân viên. Khi phát hiện lỗi, ứng dụng sẽ gửi nội dung lỗi vào màn hình và thể hiện lên cho người sử dụng biết để sửa lỗi.

### 4.3. Trang Nhân viên:

#### 4.3.1. Màn hình chính



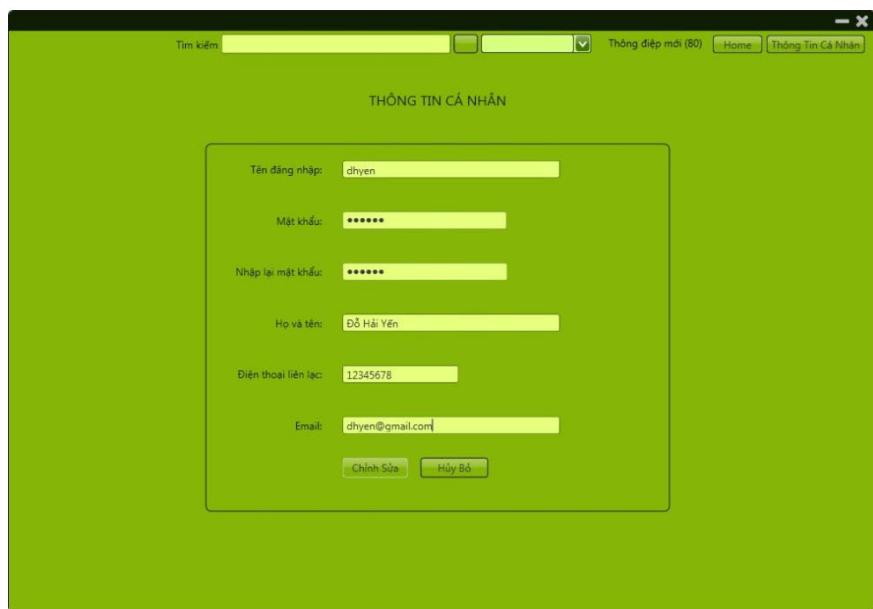
Hình 28. Màn hình chính nhân viên thura hành

STT	Nội dung	Mô tả
1	Danh sách đợt thi	Danh sách các đợt thi đang hoặc chưa thực hiện mà nhân viên có được phân công
2	Sơ đồ luồng công việc	Hiển thị luồng công việc tương ứng của quy trình. Trong đó, nhân viên chỉ có thể tương tác với các công việc dưới trách nhiệm của mình. Các tương tác bao gồm: Xem thông tin, bắt đầu công việc, chỉnh sửa hiện trạng công việc, kết thúc công việc và cập nhật báo cáo kết quả công việc.

Bảng 49. Danh sách control màn hình chính của nhân viên

#### 4.3.2. Thông tin cá nhân:

Cho phép thay đổi thông tin cá nhân người dùng



Hình 29. Màn hình cập nhật thông tin cá nhân

#### 4.4. Trang Quản Lý:

Mô tả: Màn hình chính bao gồm khung tìm kiếm và 1 menu chứa các chức năng thuộc quyền hoạt động của người quản lý, bao gồm:

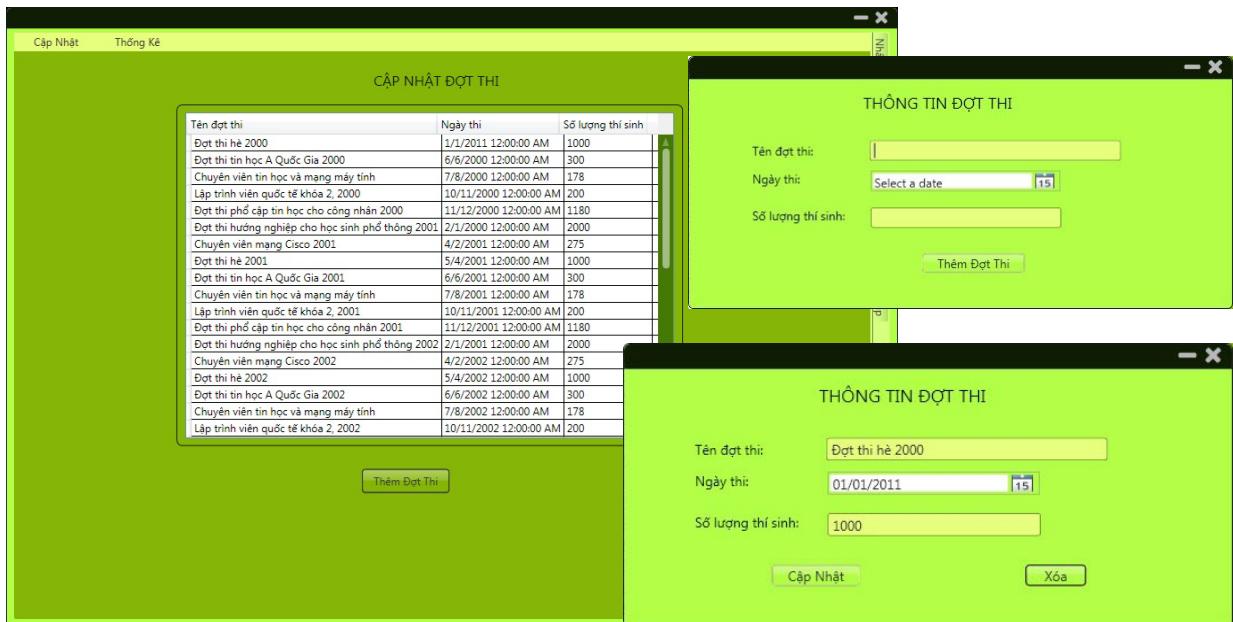
Nhân viên	Đợt thi	Hệ thống	Xem Report	Trợ giúp
Cập nhật	Cập nhật			
Phân công	Theo kì	Thống kê	Tổng quát	
	Theo đợt thi		Chi Tiết	
			So Sánh	
			Lược đồ	
			Gantt	

Hình 30. Thiết kế menu trang quản lý

Dựa vào đó, khi chọn 1 trong các chức năng trên menu, nội dung của phần làm việc tương ứng sẽ hiện ra.

#### 4.4.1. Cập nhật đợt thi:

Chức năng này cho phép người quản lý thêm và cập nhật các đợt thi trong học kì.



Hình 31. Màn hình cập nhật đợt thi

STT	Nội dung	Mô tả
1	Danh sách đợt thi	Danh sách các đợt thi có trong ứng dụng
2	Button Thêm Đợt Thi	Hiển thị màn hình thêm đợt thi

Bảng 50. Danh sách control màn hình cập nhật đợt thi

Khi người dùng chọn một đợt thi trong danh sách, màn hình cập nhật đợt thi tương ứng sẽ hiện ra bao gồm:

STT	Nội dung	Mô tả
1	Tên đợt thi	Tên đợt thi được chọn
2	Ngày thi	Ngày tổ chức thi
3	Số lượng thí sinh	Số lượng thí sinh tham gia dự thi (chỉ xem, không được sửa)
4	Button Cập nhật	Thay đổi thông tin tương ứng vào dữ liệu
5	Button Xóa	Xóa đợt thi được chọn khỏi ứng dụng

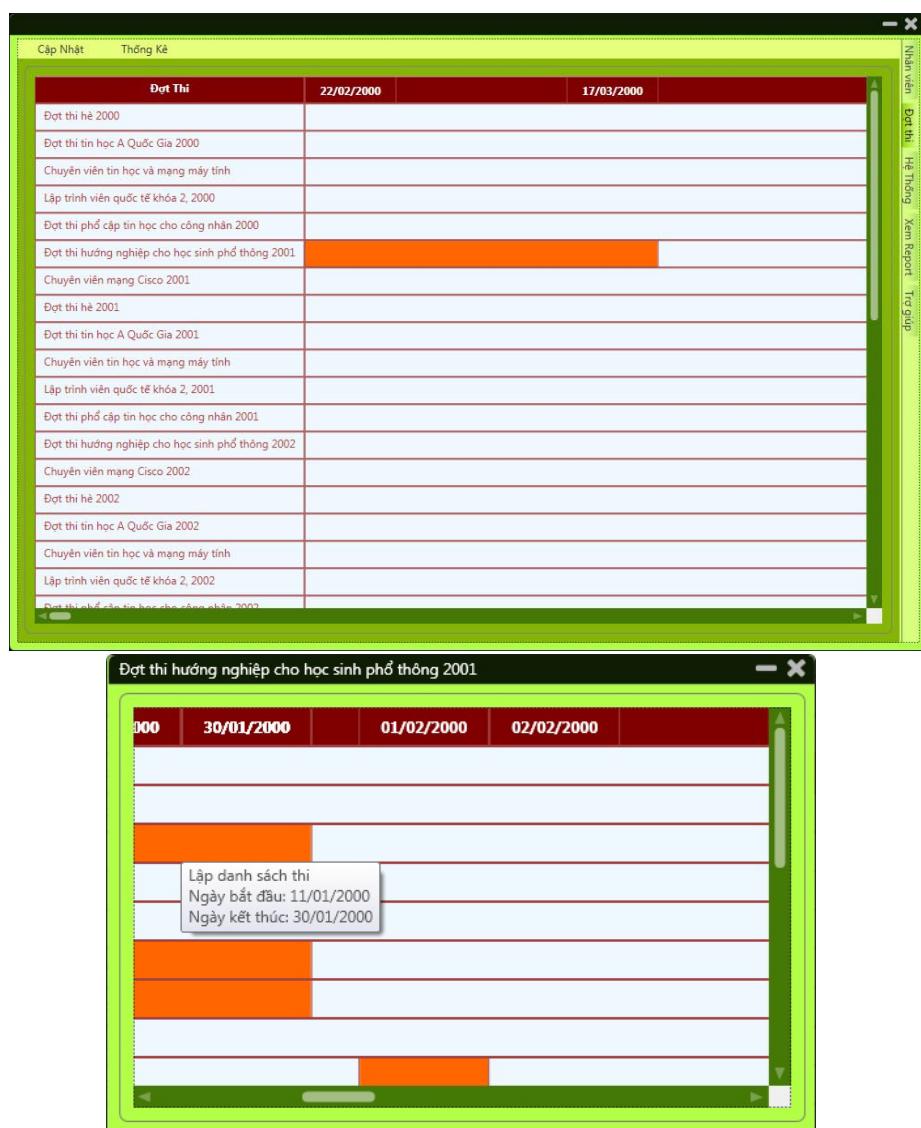
Bảng 51. Danh sách control màn hình xử lý cập nhật đợt thi

Khi người dùng chọn thêm đợt thi, màn hình thêm đợt thi sẽ hiện ra, bao gồm

STT	Nội dung	Mô tả
1	Tên đợt thi	Tên đợt thi cần thêm
2	Ngày thi	Ngày tổ chức thi
3	Số lượng thí sinh	Số lượng thí sinh tham gia dự thi: không cần nhập
4	Button Thêm đợt thi	Thêm đợt thi tương ứng vào cơ sở dữ liệu

**Bảng 52. Danh sách control màn hình thêm đợt thi**

#### 4.3.2. Thông kê Lược đồ Gantt:

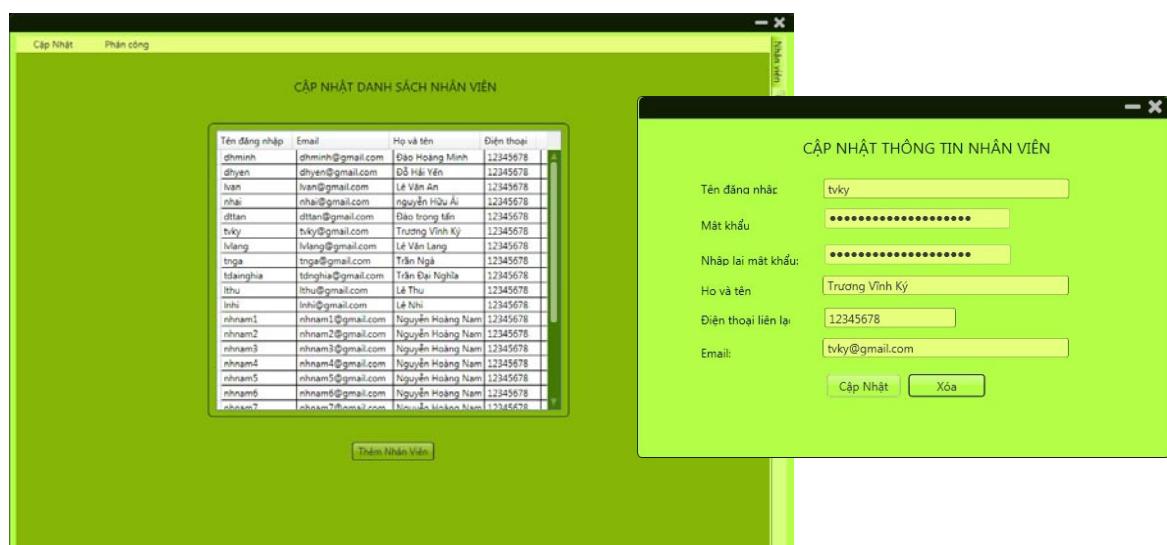


**Hình 30. Màn hình thống kê dạng lược đồ Gantt**

Hiển thị danh sách các đợt thi đang diễn ra với các thông tin tương ứng về tình trạng tiến hành. Trong đó, khi click vào 1 đợt thi tương ứng, sơ đồ gantt tương ứng của đợt thi được chọn sẽ hiện ra, thể hiện tình trạng cụ thể của từng đơn vị công việc trong đợt thi:

- Tình trạng thực thi: Đang thực thi tới gian đoạn nào
- Với mỗi giai đoạn đang/đã thực thi, phần trăm hoàn thành là bao nhiêu.
- Với những giai đoạn có nguy cơ trễ hạn, tầm quan trọng cao, ứng dụng hiển thị dưới dạng các màu khác nhau (thông nhất với các màu trên sơ đồ luồng công việc) giúp người quản lý nhanh chóng bắt được và xử lý kịp thời
- Với mỗi giai đoạn chưa được thực thi, ứng dụng sẽ hiển thị dạng mờ, tránh làm rối mắt người đọc
- Nút "Trở về" để quay lại màn hình quan sát sơ đồ gantt các đợt thi

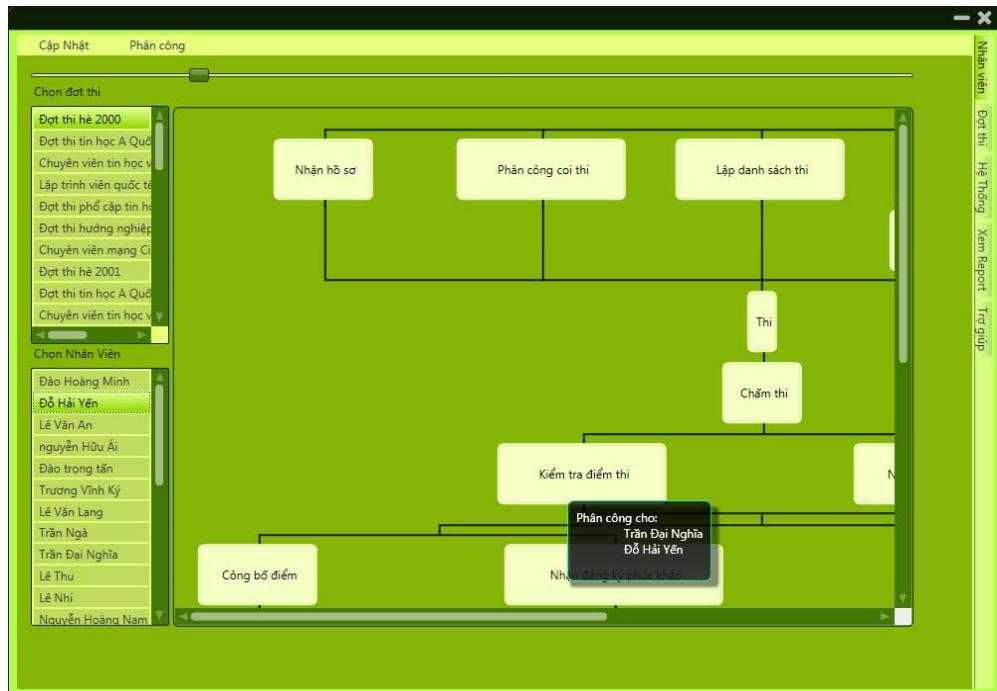
#### 4.3.3. Cập nhật nhân viên:



Hình 31. Màn hình cập nhật nhân viên

Tương tự chức năng cập nhật đợt thi.

#### 4.3.4. Phân công theo đợt thi



**Hình 32. Màn hình cập nhật phân công theo đợt thi**

STT	Nội dung	Mô tả
1	Danh sách đợt thi	Danh sách các đợt thi đang hoặc chưa được thực hiện
2	Danh sách nhân viên	Danh sách nhân viên trên ứng dụng
3	Sơ đồ luồng công việc	Sơ đồ luồng công việc tương ứng của đợt thi được chọn, với danh sách được phân công trên ứng dụng.

**Bảng 53. Danh sách control màn hình phân công theo đợt thi**

Để thay đổi, người quản lý chỉ cần chọn nhân viên cần phân công và kéo thả vào công việc tương ứng trên sơ đồ. Ứng dụng tự động cập nhật.

#### 4.3.5. Phân công theo kỳ



**Hình 33. Màn hình cập nhật phân công theo kỳ**

STT	Nội dung	Mô tả
1	Thời điểm phân công	Thời gian cần phân công công việc cho nhân viên
2	Danh sách nhân viên	Danh sách nhân viên trên ứng dụng
3	Sơ đồ luồng công việc	Sơ đồ luồng công việc tương ứng của đợt thi được chọn, với danh sách được phân công trên ứng dụng.

**Bảng 54. Danh sách control màn hình phân công theo kì**

Các thay đổi tương tự Chức năng phân công theo đợt thi. Ứng dụng tự động thay đổi phân công theo thời gian tương ứng.

#### 4.3.6. Thống kê chi tiết

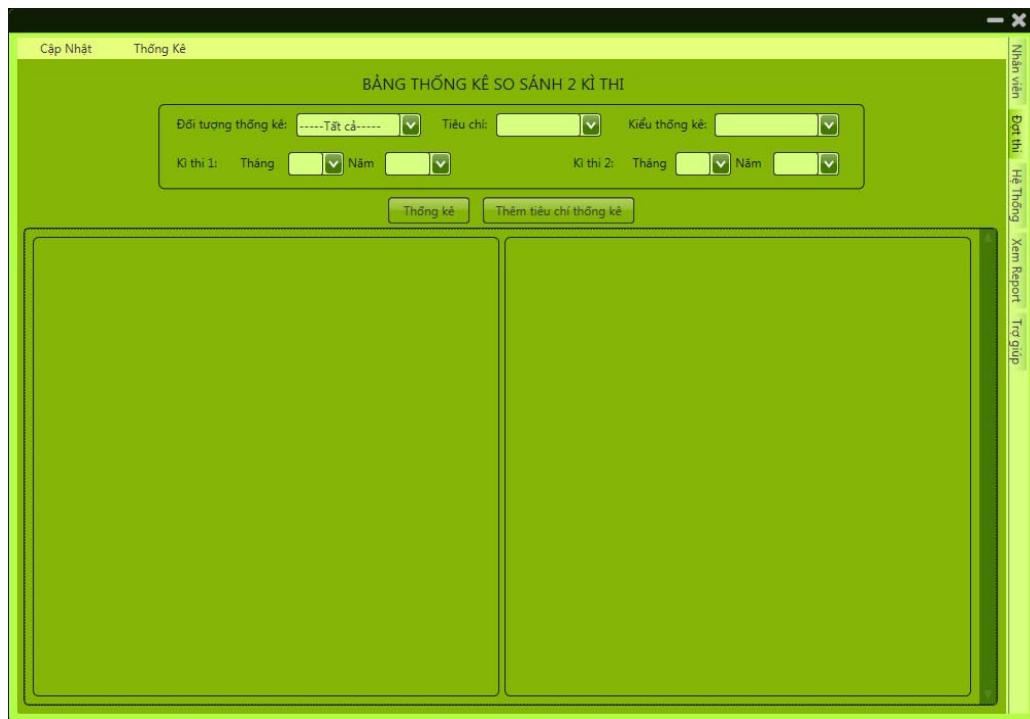


**Hình 34. Màn hình thống kê chi tiết**

STT	Nội dung	Mô tả
1	Đối tượng thống kê	Các đối tượng có liên quan trong quy trình thi, cấp chứng chỉ và in văn bằng
2	Tiêu chí	Các tiêu chí thống kê tương ứng với đối tượng thống kê
3	Kiểu thống kê	Các kiểu thống kê tương ứng với tiêu chí (Số lượng, Phần trăm, Biểu đồ...)
4	Button Thống kê	Hiển thị kết quả thống kê tương ứng bên dưới
5	Button Thêm tiêu chí	Thêm tiêu chí thống kê khác vào chương trình (xem mục i)

**Bảng 55. Danh sách control màn hình thống kê chi tiết**

#### 4.3.7. Thống kê so sánh



**Hình 35. Màn hình thống kê so sánh**

STT	Nội dung	Mô tả
1	Đối tượng thống kê	Các đối tượng có liên quan trong quy trình thi, cấp chứng chỉ và in văn bằng
2	Tiêu chí	Các tiêu chí thống kê tương ứng với đối tượng thống kê
3	Kiểu thống kê	Các kiểu thống kê tương ứng với tiêu chí (Số lượng, Phàn trǎm, Biểu đồ...)
4	Kì 1	Kì thi thứ nhất cần so sánh (tháng/năm)
5	Kì 2	Kì thi cần so sánh (tháng/năm) với kì thi thứ nhất
6	Button Thống kê	Hiển thị kết quả thống kê tương ứng bên dưới
7	Button Thêm tiêu chí	Thêm tiêu chí thống kê khác vào chương trình (xem mục i)

**Bảng 56. Danh sách control màn hình thống kê so sánh**

#### 4.3.8. Thống kê tổng hợp



**Hình 36. Màn hình thống kê tổng hợp**

STT	Nội dung	Mô tả
1	Đối tượng thống kê	Các đối tượng có liên quan trong quy trình thi, cấp chứng chỉ và in văn bằng
2	Tiêu chí	Các tiêu chí thống kê tương ứng với đối tượng thống kê
3	Kiểu thống kê	Các kiểu thống kê tương ứng với tiêu chí (Số lượng, Phần trăm, Biểu đồ...)
4	Thời gian thống kê	Khoảng thời gian cần xem thống kê
5	Button Thống kê	Hiển thị kết quả thống kê tương ứng bên dưới
6	Button Thêm tiêu chí	Thêm tiêu chí thống kê khác vào chương trình (xem mục i)

**Bảng 57. Danh sách control màn hình thống kê tổng hợp**

#### 4.3.9. Thêm tiêu chí thống kê

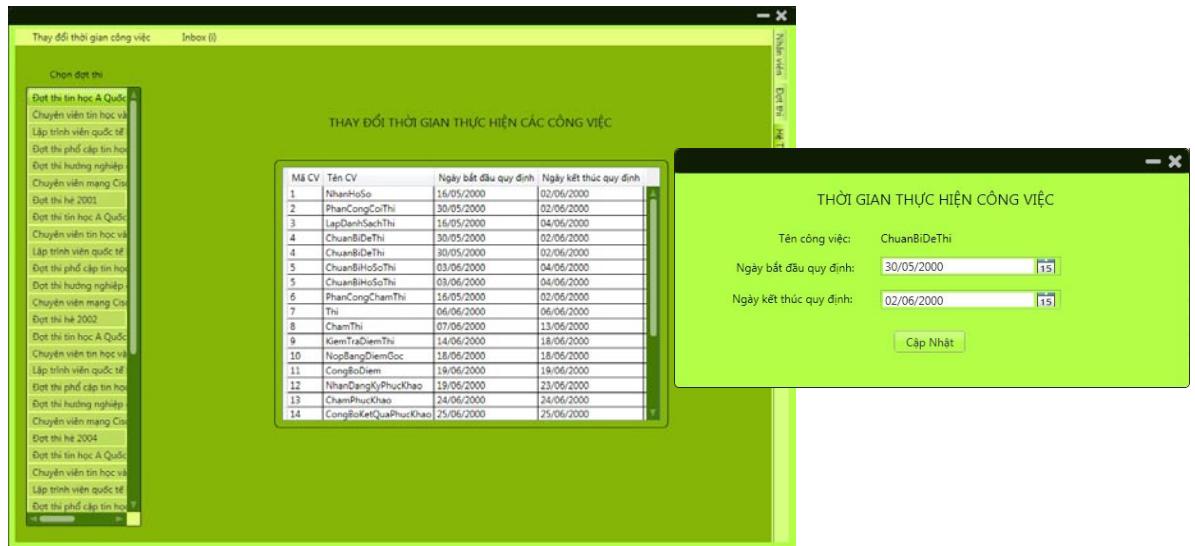


**Hình 37. Màn hình thêm tiêu chí thống kê**

STT	Nội dung	Mô tả
1	Tên tiêu chí	Tên tiêu chí thêm vào
2	Chọn bảng tham chiếu	Các bảng tương ứng trong cơ sở dữ liệu
3	Bảng sử dụng	Các bảng được chọn
4	Điều kiệu	Chọn thuộc tính, và đặt giá trị tương ứng cho thuộc tính (không hiển thị các thuộc tính bí mật)
5	Chọn cột hiển thị	Các thuộc tính sẽ hiển thị trên bảng kết quả
6	Kết quả hiển thị	Xem trước kết quả
7	Phân nhóm	Gom nhóm kết quả có thể có điều kiện
8	Thứ tự	Sắp xếp thứ tự kết quả
9	Button Thêm	Thêm Tiêu chí vào danh sách các tiêu chí thống kê
10	Button hiển thị kết quả	Chỉ hiển thị kết quả vào khung thống kê tương ứng

**Bảng 58. Danh sách control màn hình thêm tiêu chí thống kê**

#### 4.3.10. Thay đổi thời gian thực hiện công việc theo đợt thi



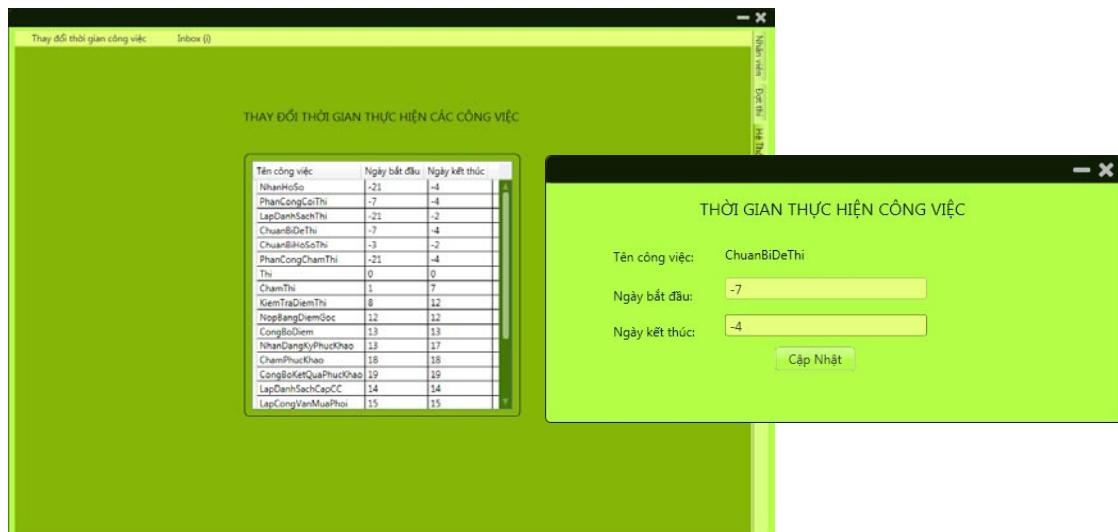
**Hình 38. Màn hình thay đổi thời gian thực hiện công việc theo đợt thi**

STT	Nội dung	Mô tả
1	Danh sách đợt thi	Các đợt thi có trong ứng dụng
2	Bảng thời gian thực hiện	Chứa các công việc với thời gian thực hiện tương ứng

**Bảng 59. Danh sách control Màn hình thay đổi thời gian thực hiện công việc theo đợt thi**

Khi quản lý chọn một công việc tương ứng trên bảng, màn hình cập nhật tương ứng hiện ra cho phép người quản lý thay đổi thời gian thực hiện quy định của công việc tương ứng

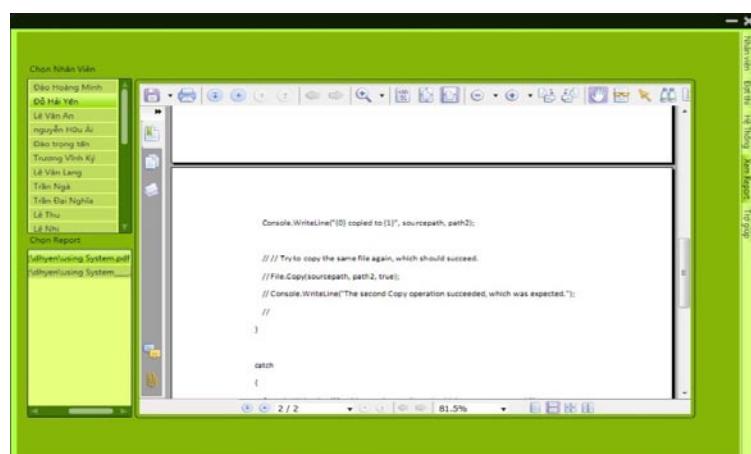
#### 4.3.11. Thay đổi thời gian thực hiện công việc



Hình 39. Màn hình thay đổi thời gian thực hiện công việc

Tương tự chức năng thực hiện công việc theo đợt thi

#### 4.3.12. Xem bản báo cáo kết quả công việc của nhân viên



Hình 40. Màn hình xem báo cáo kết quả công việc

STT	Nội dung	Mô tả
1	Danh sách nhân viên	Danh sách nhân viên trên ứng dụng
2	Danh sách báo cáo	Các báo cáo mà nhân viên tương ứng được chọn đã cung cấp lên ứng dụng
3	Bảng báo cáo	Xem bảng báo cáo tương ứng được chọn

Bảng 60. Danh sách control Màn hình xem báo cáo kết quả công việc