

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

**XÂY DỰNG HỆ THỐNG QUẢN LÝ CÔNG
VIỆC TRỰC TUYẾN ETASK**

Sinh viên thực hiện : **Nguyễn Tuấn Anh**

Lớp KTMT – K51

Giáo viên hướng dẫn: ThS **Nguyễn Tử Quảng**

HÀ NỘI 5-2011

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Tuấn Anh

Điện thoại liên lạc: 01656112624

Email: anhnt.ktmt@gmail.com

Lớp: Kỹ thuật máy tính K51

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Công ty an ninh mạng BKAV

Thời gian làm ĐATN: Từ ngày 20 /02 /2011 đến 27 /05 /2011

2. Mục đích nội dung của ĐATN

Phát triển phần lập lịch làm việc trong hệ thống quản lý công việc trực tuyến eTask.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu về mô hình client-server.
- Tìm hiểu giao thức và thiết kế giao thức cho phần client và server.
- Tìm hiểu về phương pháp phân tích thiết kế phần mềm hướng đối tượng.
- Phân tích các yêu cầu của module lập lịch làm việc.
- Thiết kế chi tiết module lập lịch thỏa mãn các yêu cầu về lập và quản lý lịch đa người dùng.
- Cài đặt bản phần mềm bằng ngôn ngữ lập trình.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Tuấn Anh* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS Nguyễn Tử Quảng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 26 tháng 05 năm 2011

Tác giả ĐATN

Nguyễn Tuấn Anh

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

Hà Nội, ngày tháng năm 2011

Giáo viên hướng dẫn

ThS Nguyễn Tử Quảng

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Đồ án tập trung vào phân tích yêu cầu, thiết kế module lập lịch làm việc của “hệ thống quản lý công việc trực tuyến eTask” và nghiên cứu các công nghệ liên quan. Đồ án được chia làm 3 phần với nội dung tóm tắt như sau:

Phần 1:Đặt vấn đề và định hướng giải pháp.

- Trình bày về bối cảnh ra đời, nhu cầu thực tiễn và đánh giá, mục đích và nhiệm vụ của đề tài.
- Một số kiến thức liên quan tới việc xây dựng và phát triển hệ thống như: lý thuyết về mô hình client-server, phương pháp phát triển hệ thống và UML.

Phần 2: Các kết quả đạt được.

Phần này trình bày kết quả của việc khảo sát, phân tích, thiết kế và xây dựng hệ thống:

- Khảo sát và phân tích các yêu cầu của hệ thống, đặc tả các chức năng của hệ thống bằng các biểu đồ ca sử dụng.
- Mô hình hóa cấu trúc tĩnh của hệ thống bằng các biểu đồ lớp, biểu đồ lớp tham gia các ca sử dụng.
- Mô hình hóa khía cạnh tĩnh của hệ thống bằng các biểu đồ trình tự.
- Mô tả giao diện hệ thống bằng nguyên mẫu giao diện.
- Thiết kế kiến trúc tổng quan của hệ thống.
- Thiết kế chi tiết hệ thống, kết quả là biểu đồ lớp thiết kế.

Phần kết luận:

Trong phần này sẽ trình bày về các kết quả đã đạt được, chưa đạt được và định hướng phát triển hệ thống trong tương lai.

ABSTRACT OF THESIS

The thesis focuses on the analysis and design of the software that helps to manage the work. The thesis is divided into three parts, as follows:

Part 1: Introduction

- Presenting background, reality demands, tasks, purposes, and scope of the thesis.
- Presenting related knowledge, and how the knowledge used to build the software as: client-server models, network theories, multithread programming techniques, Object-Oriented System Development and UML.

Part 2: Presenting achievement of software investigation, analysis and designation

- Investigating and analyzing requirements of the software, specifying functions of the software.
- Modeling system by UML charts.
- Using interface prototype to describes system interface.
- General system design

Part 3: Conclusion

Presenting real result, achievement and shortcoming, development direction in the future.

LỜI CẢM ƠN

Để có thể hoàn thành đồ án tốt nghiệp này, em xin chân thành cảm ơn các thầy cô trong trường Đại học Bách Khoa Hà Nội và đặc biệt là các thầy cô trong Viện Công nghệ thông tin & Truyền thông đã tận tình dạy dỗ em trong suốt những năm học vừa qua. Em xin cảm ơn Công ty TNHH an ninh mạng Bkav đã tạo điều kiện cho em có một môi trường thực tập và làm việc gần một năm qua. Em xin gửi lời cảm ơn tới gia đình, bạn bè, các anh chị trong công ty Bkav đã giúp đỡ động viên em trong quá trình học tập cũng như làm đồ án.

Cuối cùng em gửi lời cảm ơn sâu sắc tới thầy Nguyễn Tử Quảng đã tận tình chỉ dẫn và tạo điều kiện cho em hoàn thành đồ án này.

Hà nội ngày 26/05/2011

Sinh viên

Nguyễn Tuấn Anh

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	1
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP	2
ABSTRACT OF THESIS	3
DANH MỤC CÁC BẢNG.....	6
DANH MỤC CÁC HÌNH.....	7
DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ	8
PHẦN I. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP	9
CHƯƠNG 1. ĐẶT VẤN ĐỀ.....	9
1.1 Bối cảnh.....	9
1.2 Nhu cầu thực tế	9
1.3 Nhiệm vụ của đồ án.....	10
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	11
2.1 Mô hình client – server	11
2.2 Phát triển hệ thống hướng đối tượng và UML	12
PHẦN II. CÁC KẾT QUẢ ĐẠT ĐƯỢC.....	20
CHƯƠNG 3. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG	20
3.1 Các yêu cầu của hệ thống.....	20
3.2 Nhận định và đặc tả các ca sử dụng	22
3.3 Mô hình hóa cấu trúc.....	31
3.4 Mô hình hóa hành vi	36
CHƯƠNG 4. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG	44
4.1 Làm nguyên mẫu giao diện người dùng.....	44
4.2 Thiết kế hệ thống.....	51
4.3 Thiết kế chi tiết.....	51
4.4 Kết quả đạt được	56
PHẦN KẾT LUẬN	58
CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐỊNH HƯỚNG PHÁT TRIỂN	58
5.1 Kết quả đạt được	58
5.2 Định hướng phát triển	58
TÀI LIỆU THAM KHẢO.....	59
PHỤ LỤC.....	60
1 Các nguyên mẫu giao diện của ứng dụng	60

DANH MỤC CÁC BẢNG

<i>Bảng 1. Bảng các thuật ngữ.....</i>	<i>8</i>
<i>Bảng 2. Bảng từ viết tắt.....</i>	<i>8</i>
<i>Bảng 3. Mô tả giao diện: Giao diện chính.....</i>	<i>44</i>
<i>Bảng 4. Phân bố các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’</i>	<i>46</i>
<i>Bảng 5. Mô tả giao diện: Cửa sổ tạo lịch mới.....</i>	<i>47</i>
<i>Bảng 6. Mô tả giao diện: Cửa sổ xung đột lịch</i>	<i>47</i>
<i>Bảng 7. Mô tả giao diện: Cửa sổ xem lịch.....</i>	<i>48</i>
<i>Bảng 8. Mô tả giao diện: Cửa sổ thống kê lịch</i>	<i>48</i>
<i>Bảng 9. Phân bố các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’</i>	<i>49</i>
<i>Bảng 10. Mô tả giao diện: Cửa sổ tìm kiếm lịch</i>	<i>49</i>
<i>Bảng 11. Mô tả giao diện: Cửa sổ tìm thời gian trống.....</i>	<i>50</i>
<i>Bảng 12. Phân bố các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’</i>	<i>50</i>
<i>Bảng 13. Mô tả giao diện: Cửa sổ tìm kiếm lịch</i>	<i>50</i>

DANH MỤC CÁC HÌNH

Hình 1. Mô hình IOCP	12
Hình 2. Các góc nhìn trong UML	13
Hình 3. Phân loại các biểu đồ trong UML 2.0	15
Hình 4. Các pha của RUP	16
Hình 5. Tiến trình mười bước	17
Hình 6. Biểu đồ ca sử dụng tổng quan	22
Hình 7. Biểu đồ ca sử dụng quản lý lịch	23
Hình 8. Biểu đồ ca sử dụng thống kê lịch	25
Hình 9. Biểu đồ ca sử dụng tìm kiếm lịch	26
Hình 10. Biểu đồ ca sử dụng duyệt tài nguyên	27
Hình 11. Biểu đồ ca sử dụng quản lý tài nguyên	28
Hình 12. Biểu đồ ca sử dụng cấu hình phân lập lịch	30
Hình 13. Biểu đồ lớp lĩnh vực	32
Hình 14. Biểu đồ lớp tham gia của ca sử dụng 'Quản lý lịch'	33
Hình 15. Biểu đồ lớp tham gia của ca sử dụng 'Thống kê lịch'	34
Hình 16. Biểu đồ lớp tham gia của ca sử dụng 'Tìm kiếm lịch'	34
Hình 17. Biểu đồ lớp tham gia của ca sử dụng 'Duyệt tài nguyên'	35
Hình 18. Biểu đồ lớp tham gia của ca sử dụng 'Quản lý tài nguyên'	35
Hình 19. Biểu đồ trình tự cho "Quản lý lịch/Sửa, Xóa lịch"	37
Hình 20. Biểu đồ trình tự cho "Quản lý lịch/Tạo lịch mới"	38
Hình 21. Biểu đồ trình tự cho "Thống kê lịch"	39
Hình 22. Biểu đồ trình tự cho "Tìm kiếm lịch/tìm lịch theo thông tin lịch"	39
Hình 23. Biểu đồ trình tự cho "Tìm kiếm lịch/Tìm thời gian trống"	40
Hình 24. Biểu đồ trình tự cho "Duyệt tài nguyên"	41
Hình 25. Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý tài nguyên"	42
Hình 26. Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý node"	43
Hình 27. Nguyên mẫu giao diện chính (xem theo tuần)	45
Hình 28. Nguyên mẫu giao diện xem lịch theo ngày	45
Hình 29. Nguyên mẫu giao diện xem lịch theo tháng	46
Hình 30. Mô hình bố trí vật lý của hệ thống	51
Hình 31. Biểu đồ lớp thiết kế thuộc gói lập lịch	53
Hình 32. Biểu đồ lớp thiết kế thuộc gói Socket	54
Hình 33. Biểu đồ lớp thiết kế thuộc gói 'Thao tác dữ liệu'	54
Hình 34. Biểu đồ lớp thiết kế thuộc gói server	55
Hình 35. Giao diện chính	56
Hình 36. Cửa sổ tạo lịch mới	57
Hình 37. Màn hình duyệt tài nguyên	60
Hình 38. Trình bày thông tin một lịch trên các view	60
Hình 39. Tip của lịch (hiển thị khi di chuột qua lịch)	60
Hình 40. Cửa sổ tạo lịch mới	61
Hình 41. Tab chi tiết của cửa sổ tạo lịch	61

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

THUẬT NGỮ

Thuật ngữ	Ý nghĩa
Lịch	
Tài nguyên	Là những vật dụng, thiết bị, văn phòng, thời gian liên quan tới lịch: <ul style="list-style-type: none"> - Vật dụng, thiết bị: máy tính, máy chiếu, phương tiện - Thời gian: thời gian nghỉ phép, thời gian trực của nhân viên. - Văn phòng: phòng họp, phòng ốc...
Xung đột	Trong một khoảng thời gian một người chỉ có thể tham gia vào một lịch, vì vậy khi đã tham gia vào một lịch mà có một lịch khác được tạo với họ sẽ gây ra xung đột.
Client	Máy khách
Server	Máy chủ

Bảng 1. Bảng các thuật ngữ

TỪ VIẾT TẮT

Từ viết tắt	Viết đầy đủ
ĐATN	Đồ án tốt nghiệp
VC++	Visual C++
IOCP	Input/Output Completion Port
MFC	Microsoft Foundation Classes
SQL	Structured Query Language
RUP	Rational Unified Process

Bảng 2. Bảng từ viết tắt

PHẦN I. ĐẶT VẤN ĐỀ VÀ ĐỊNH HƯỚNG GIẢI PHÁP

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1 Bối cảnh

Những năm gần đây, bằng cách hỗ trợ các ngành khác, công nghệ thông tin (CNTT) đang là một ngành thúc đẩy sự phát triển của nền kinh tế.

Tuy nhiên, việc ứng dụng CNTT trong các công ty, tổ chức, doanh nghiệp trong nước hiện nay vẫn còn nhiều hạn chế. Một phần là do rào cản về ngôn ngữ (hầu hết các phần mềm có ngôn ngữ tiếng Anh), một phần là do các phần mềm này chưa phù hợp với thói quen của người của người sử dụng.

1.2 Nhu cầu thực tế

Trong hầu hết các doanh nghiệp, cơ quan, tổ chức đều có trang bị máy tính cho nhân viên và hệ thống mạng nội bộ kết nối các máy tính này. Nhiều công việc văn phòng đã được tin học hóa thông qua việc sử dụng các phần mềm hỗ trợ, nhờ đó các công việc được tiến hành nhanh chóng hiệu quả hơn.

Trong một cơ quan thì việc họp diễn ra rất thường xuyên. Để có một lịch họp với cấp trên, với những người liên quan tới công việc của mình, công việc của người đặt lịch thường là: hỏi từng người mà mình cần họp xem họ có bận gì khi đó không, tiếp đến là kiểm tra xem phòng họp nào còn trống, lên lịch họp cuối cùng là thông báo lại với từng người lịch họp đó. Các lịch cơ quan, lịch trực thường được bộ phận văn phòng sắp xếp, rồi dán trên bảng tin cho mọi người. Như vậy để có được một lịch họp mất rất nhiều thời gian, công sức và những người liên quan đến lịch khó theo dõi lịch của mình.

Để đáp ứng nhu cầu này hiện nay trên thị trường xuất hiện một số phần mềm phục vụ cho việc quản lý lịch làm việc trên nền web và nền desktop. Phần mềm trên desktop có thể kể đến: MS Outlook, Task Manager, Mozilla Thunderbird,..., phần mềm trên nền web có: eGroupWare, Google Calendar,...Nhưng các phần mềm này bộc lộ một số hạn chế khiến cho việc sử dụng gặp khó khăn như:

- Hầu hết các phần mềm hiện nay đang được sử dụng cho chức năng tạo và quản lý lịch làm việc là phần mềm nước ngoài, với giao diện tiếng Anh gây khó khăn trong việc sử dụng và khai thác hết tính năng của phần mềm.
- Phần mềm trên nền web có ưu điểm là có các tính năng quản lý lịch khá đầy đủ, hỗ trợ đa người dùng, nhưng tốc độ chậm, việc tương tác với người dùng kém linh hoạt.
- Phần mềm trên nền windows thì ít tính năng hơn (chưa hỗ trợ đa người dùng), nhưng tốc độ xử lý nhanh, tương tác với người dùng linh hoạt hơn.

Từ hiện trạng đó, việc xây dựng một phần mềm với giao diện hoàn toàn tiếng Việt có hỗ trợ chức năng tạo lịch, quản lý lịch đa người dùng, dễ sử dụng, tương tác linh hoạt là thực sự cần thiết.

1.3 Nhiệm vụ của đồ án

Với thực tiễn trên, bộ phận phần mềm công ty an ninh mạng Bkav đang xây dựng “hệ thống quản lý công việc trực tuyến eTask”. Em tham gia phát triển module “lập lịch làm việc” cho hệ thống này. Mục đích của đồ án là phát triển module lập lịch, có hỗ trợ chức năng lập lịch làm việc qua mạng LAN. Đồ án phải đáp ứng được các yêu cầu sau:

➤ Về lý thuyết:

- Tìm hiểu về phát triển hệ thống hướng đối tượng, ngôn ngữ mô hình hóa UML.
- Tìm hiểu các kiến thức về lập trình mạng, các mô hình client-server, giao thức mạng.
- Tìm hiểu về đa luồng, cách xử lý xung đột trong lập trình đa luồng.
- Đưa ra được bản phân tích và thiết kế chi tiết module lập lịch làm việc, để đưa vào xây dựng hệ thống.

➤ Về sản phẩm:

- Module phải đáp ứng được các chức năng cơ bản của phần mềm quản lý lịch làm việc cá nhân, lịch làm việc của nhóm người.
- Hoạt động tốt trên môi trường mạng LAN với nhiều người làm việc cùng nhau.
- Chương trình bố trí giao diện và chức năng hợp lý để dùng ngay cả với những người có ít kiến thức về công nghệ thông tin.

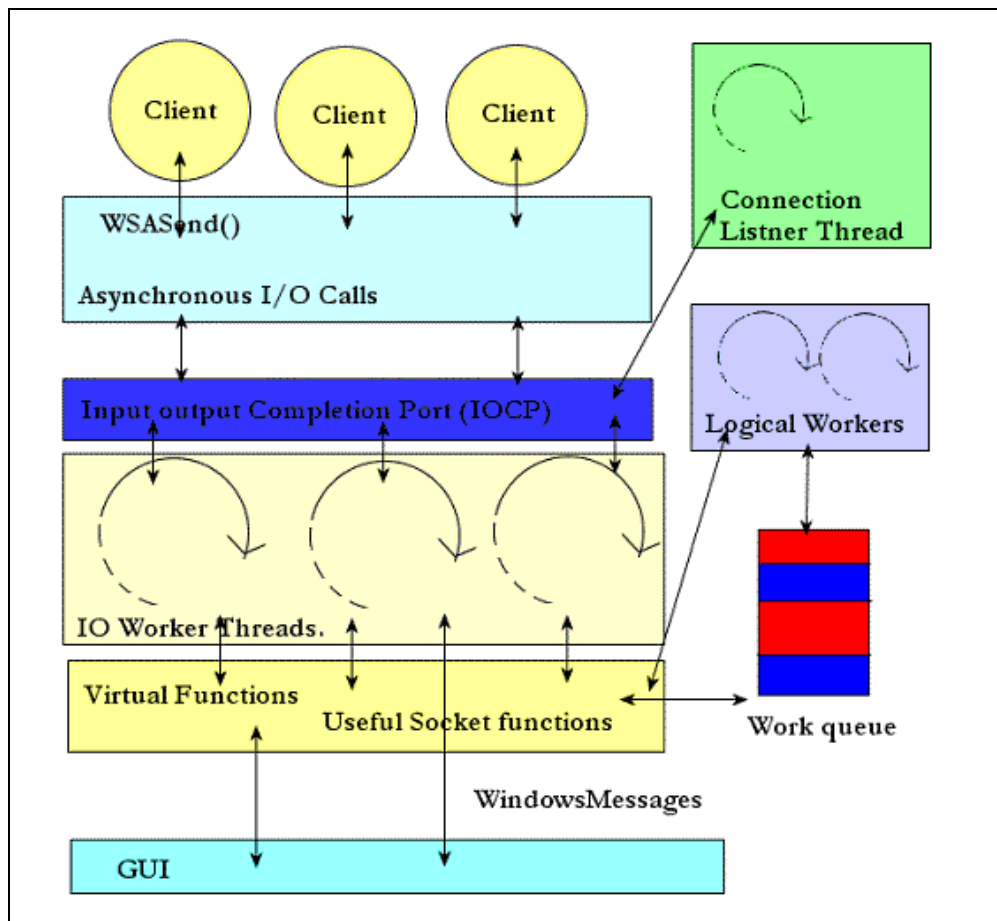
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Mô hình client – server

Mô hình client-server là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (request) đến máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

Khi tổ chức phần mềm dạng client-server thì việc lựa chọn mô đáp ứng của server là điều rất quan trọng vì nó giúp giảm tải độ phức tạp trong xây dựng hệ thống, cũng như đảm bảo các yêu cầu về hiệu năng của hệ thống. Có một số mô hình đáp ứng được như sau:

- **Mô hình một tiến trình:** Quá trình thực hiện trao đổi giữa các client và server thực hiện đồng thời trên một tiến trình chính. Ưu điểm của mô hình này là việc xử lý tập trung và đơn giản. Nhược điểm của nó là khi có nhiều kết nối tới server thì việc tạo ra nhiều liên kết sẽ gây khó khăn cho quá trình quản lý kết nối, và quá trình trao đổi sẽ bị chậm đi do bị chia sẻ.
- **Mô hình đa tiến trình:** Mỗi client được phục vụ trong 1 tiến trình. Mô hình này sẽ làm giảm mức độ trao đổi của chương trình chính đáng kể vì mỗi kết nối đều được cung cấp 1 tiến trình để trao đổi riêng. Nhưng khi có nhiều kết nối thì việc tạo ra nhiều tiến trình như vậy sẽ tiềm ẩn nguy cơ về độ bền tiến trình, đồng thời khó kiểm soát lỗi.
- **Mô hình đa tiến trình IOCP:** Mô hình IOCP là kỹ thuật được đưa ra nhằm giải quyết vào ra không đồng bộ. Bằng việc sử dụng một số lượng nhỏ các thread theo một phương thức hiệu quả.
 - Khi một tiến trình tạo ra một cổng vào ra đầy đủ, hệ thống tạo ra một đối tượng danh sách liên kết cho các yêu cầu và một tập các luồng mà mục đích duy nhất của nó là để phục vụ cho các yêu cầu này.
 - Các tiến trình mà có nhiều yêu cầu vào ra không đồng bộ đồng thời có thể thực hiện nhanh hơn nhờ IOCP thay vì tại mỗi thời điểm nhận được yêu cầu nó lại sinh ra một luồng để xử lý yêu cầu đó.
 - Khi một cổng vào ra đầy đủ thực hiện xong một vào ra ngay lập tức nó sẽ kiểm tra xem có yêu cầu nào trong hàng đợi hay không, nếu tồn tại nó sẽ lấy các yêu cầu đang chờ trong hàng đợi vào để xử lý, ngược lại quá trình kiểm tra sẽ tiếp tục.



Hình 1. Mô hình IOCP

2.2 Phát triển hệ thống hướng đối tượng và UML

Phát triển hệ thống được hiểu là quá trình xây dựng một hệ thống tin học, kể từ lúc manh nha ý đồ, đến khảo sát để tìm hiểu môi trường nhu cầu, rồi phân tích để đi sâu vào chi tiết, thiết kế để làm cho nó thích ứng với điều kiện kỹ thuật sẵn có, cài đặt để thực thi nó trong một ngôn ngữ lập trình và trên một nền tảng kỹ thuật, và cuối cùng là kiểm chứng và chuyển giao.

Tồn tại khá nhiều phương pháp để phát triển hệ thống. Tuy có nhiều phương pháp, song một cách đại thể, có thể tách chúng thành hai nhóm lớn tùy vào hai định hướng khác biệt. Đó là phương pháp hướng chức năng và phương pháp hướng đối tượng.

Các phương pháp hướng chức năng, nở rộ vào những năm 70, 80 của thế kỷ trước, lấy chức năng làm đơn vị phân rã khi tiến hành phân tích hệ thống. Câu hỏi về hệ thống thường được đặt ra sớm nhất cho người dùng, cũng như cho người thiết kế là câu hỏi: “Hệ thống làm gì?”. Bởi vậy nghiên cứu hệ thống dựa vào các chức năng (tức là việc phải làm) là một cách tự nhiên và dễ hiểu. Phương pháp hướng chức năng sẽ dẫn tới việc cài đặt hệ thống bằng các ngôn ngữ lập trình thủ tục (như Pascal, C...). Dù dễ làm, dễ hiểu nhưng nó có các nhược điểm khó chấp nhận: đó là

các hệ thống xây dựng theo phương pháp này là khó sửa chữa, khó nâng cấp và ít có khả năng tái sử dụng vào các hoàn cảnh khác.

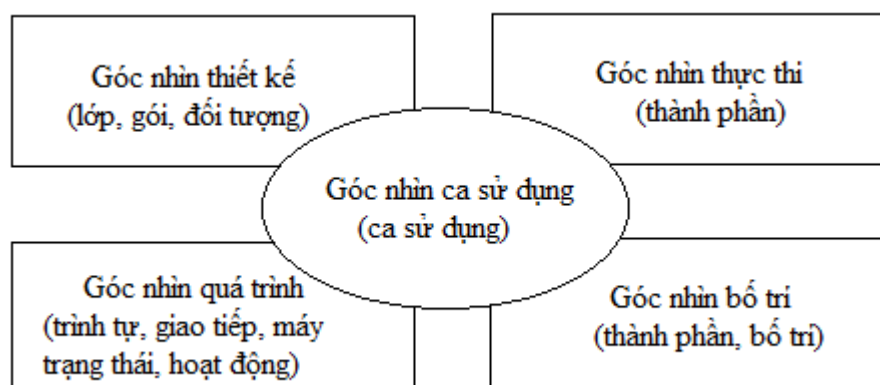
Các phương pháp hướng đối tượng khắc phục được các nhược điểm trên và ra đời từ đầu các năm 90 đến nay, lại lấy đối tượng làm nguyên đơn cơ bản của hệ thống. Đối tượng là một sự kết hợp giữa các chức năng và dữ liệu. Đó là một sự kết hợp hợp lý, vì mỗi chức năng chỉ thao tác trên một số dữ liệu nhất định. Không những hợp lý mà lại còn rất tự nhiên và dễ hiểu, vì các đối tượng tin học thường dùng để phản ánh hay mô phỏng các đối tượng trong thế giới thực.

Dù là dùng phương pháp nào, thì người phân tích và thiết kế luôn phải dùng một hình thức hiểu được nào đó để diễn tả các sắc thái khác nhau của hệ thống. Hình thức diễn tả đó có thể ở dạng văn tự, các bảng hoặc các đồ thị. Người ta gọi đó là các mô hình, và việc mô hình để diễn tả hệ thống được gọi là mô hình hóa. Trước đây mỗi phương pháp phát triển hệ thống đề nghị một loại mô hình riêng. Sự khác biệt trong ngôn ngữ diễn tả hệ thống hệ thống đó, giống như người nói tiếng Anh, kẻ nói tiếng Việt về cùng một vấn đề, đã gây ra những khó khăn không cần thiết. Xu hướng tất yếu là phải đi đến thống nhất về ngôn ngữ. Vì vậy mà ngôn ngữ mô hình hóa thống nhất UML ra đời.

2.2.1 Ngôn ngữ UML

a) Các góc nhìn của UML

UML cung cấp các mô hình để diễn tả hệ thống. Nhưng mỗi mô hình thì chỉ có thể diễn tả hệ thống theo một góc nhìn nhất định. Tổng hợp lại thì UML cung cấp năm góc nhìn đối với hệ thống. Mỗi góc nhìn thực hiện bởi một số biểu đồ. Có thể có biểu đồ cùng thuộc vào các góc nhìn khác nhau.



Hình 2. Các góc nhìn trong UML

Góc nhìn ca sử dụng: Là góc nhìn từ ngoài vào hệ thống. Đó là cách nhìn của các người dùng cuối, các người phân tích, người kiểm định. Nó không phản ánh tổ chức bên trong của phần mềm, mà chỉ làm rõ các chức năng lớn mà hệ thống phải đáp ứng cho người dùng. Với UML thì sắc thái tĩnh của góc nhìn này được thu tóm

trong biểu đồ ca sử dụng, còn góc sắc thái động được thu tóm trong các biểu đồ tương tác, các biểu đồ máy trạng thái và các biểu đồ hoạt động.

Góc nhìn thiết kế (còn gọi là góc nhìn logic): Là góc nhìn vào bên trong hệ thống, cho thấy các nhiệm vụ của hệ thống được thiết kế ra sao (thành phần các lớp, các giao diện, các hợp tác). Đó là cách nhìn của các người thiết kế hệ thống. Với UML, thì sắc thái tĩnh của góc nhìn này thể hiện trong các biểu đồ lớp, các biểu đồ đối tượng. Còn sắc thái động của góc nhìn này thể hiện trong các biểu đồ tương tác, các biểu đồ máy trạng thái, các biểu đồ hoạt động.

Góc nhìn quá trình (còn gọi là góc nhìn song hành): Phản ánh các lộ trình điều khiển, các quá trình thực hiện, cho thấy sự hoạt động song hành hay đồng bộ của hệ thống. Với UML thì góc nhìn này được thể hiện cùng với các biểu đồ như góc nhìn thiết kế, nhưng tập trung vào các lớp chủ động, là các lớp biểu diễn cho các lộ trình điều khiển và quá trình thực hiện.

Góc nhìn thực thi (còn gọi là góc nhìn thành phần): Là góc nhìn đối với dạng phát hành của phần mềm bao gồm các thành phần và tệp tương đối độc lập có thể được lắp ráp theo nhiều cách để tạo ra hệ thống chạy được. Với UML, sắc thái tĩnh của góc nhìn này được thể hiện bởi các biểu đồ thành phần. Còn sắc thái động của góc nhìn này được thể hiện bởi các biểu đồ tương tác, biểu đồ máy trạng thái, các biểu đồ hoạt động.

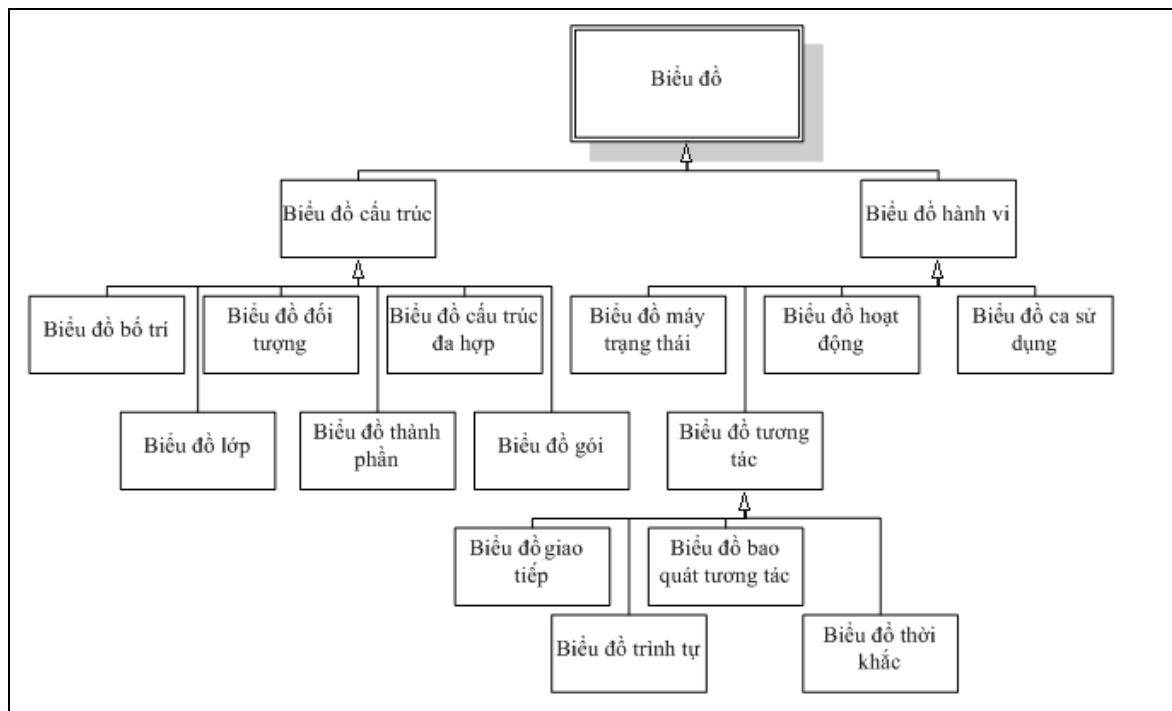
Góc nhìn bố trí: Là góc nhìn về tập ô của phần cứng mà trên đó hệ thống được thực hiện. Nó chỉ rõ sự phân bố, sự sắp đặt các thành phần của hệ thống vật lý trên các đơn vị phần cứng. Với UML thì sắc thái tĩnh của góc nhìn này thể hiện qua các biểu đồ bố trí. Còn sắc thái động của góc nhìn này thể hiện qua các biểu đồ tương tác, các biểu đồ máy trạng thái, các biểu đồ hoạt động.

b) Các biểu đồ của UML

Các mô hình trong UML đều có các dạng biểu đồ. Biểu đồ là một đồ thị, trong đó:

- Các nút (nodes) là các yếu tố của mô hình có dạng đồ họa hai chiều, như lớp, trạng thái, gói...
- Các đường (paths) là các yếu tố mô hình có dạng đồ hoặc tuyến tính, như liên kết, khái quát hóa, phụ thuộc...

Trong UML 2.0 có 13 loại biểu đồ khác nhau. Chúng được phân thành hai nhóm: các biểu đồ về cấu trúc và các biểu đồ về hành vi.



Hình 3. Phân loại các biểu đồ trong UML 2.0

2.2.2 Tiến trình RUP

Tiến trình RUP (Rational Unified Process) là một tiến trình mô hình hóa với UML, song nó không phải là chuẩn.

a) Các nguyên tắc cơ bản của RUP

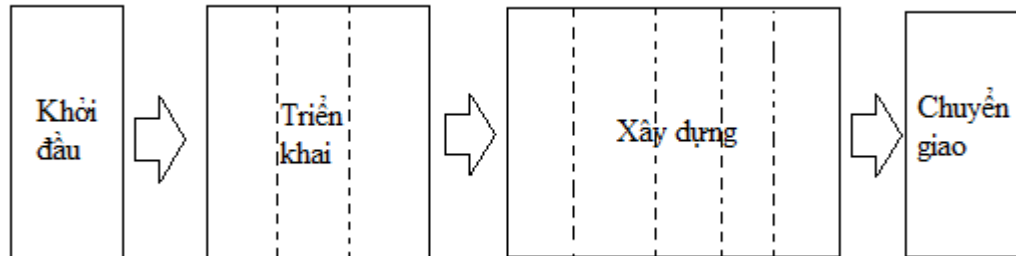
Tiến trình RUP là một tiến trình phát triển phần mềm dựa trên các nguyên tắc: “lặp và tăng trưởng, tập trung vào kiến trúc, dẫn dắt theo các ca sử dụng và không chế bởi các nguy cơ”.

- Lặp và tăng trưởng: Dự án được cắt thành những vòng lặp hoặc giai đoạn ngắn (khoảng một tháng) cho phép kiểm soát dễ dàng sự tiến triển của dự án. Cuối mỗi vòng lặp thì một phần thi hành được của hệ thống được sản sinh theo cách tăng trưởng (thêm vào) dần dần.
- Tập trung vào kiến trúc: Toàn bộ hệ thống phức tạp được phân chia thành từng phần để có thể dễ dàng triển khai và duy tu, tạo nên một kiến trúc. Kiến trúc này phải được trình bày theo năm góc nhìn khác nhau bởi các biểu đồ của UML.
- Dẫn dắt theo ca sử dụng: RUP nhấn mạnh sự đáp ứng các nhu cầu của người dùng, thể hiện bởi các ca sử dụng. Do đó các ca sử dụng ảnh hưởng và dẫn đường cho mọi giai đoạn phát triển hệ thống. Nắm bắt nhu cầu là để phát hiện các ca sử dụng. Phân tích là đi sâu vào các ca sử dụng. Thiết kế và cài đặt là để xây dựng hệ thống theo từng ca sử dụng. Kiểm định và nhiệm vụ hệ thống thực hiện theo các ca sử dụng. Ca sử dụng là căn cứ để xác định các vòng lặp và tăng trưởng và cũng là căn cứ để phân công công việc trong nhóm phát triển hệ thống.

- Không chế bởi các nguy cơ: Các nguy cơ chính đối với dự án phải phát hiện sớm và loại bỏ càng sớm càng tốt. Yêu cầu này là căn cứ để xác định thứ tự trước sau của vòng lặp.

b) Các pha và công đoạn của tiến trình RUP

Tiến trình RUP được tổ chức thành bốn pha nối tiếp trong thời gian, là: khởi đầu, triển khai, xây dựng và chuyển giao.



Hình 4. Các pha của RUP

- Pha khởi đầu: nhằm cho một cái nhìn tổng quát về hệ thống sẽ xây dựng (chức năng, hiệu năng, công nghệ, ...) và về dự án sẽ triển khai (phạm vi, mục tiêu, tính khả thi, ...). Từ đó đưa ra kết luận là nên phát triển hay nên loại bỏ dự án.
- Pha triển khai: bao gồm sự phân chia chi tiết hơn về hệ thống, cả về chức năng và cấu trúc tĩnh (dùng các biểu đồ ca sử dụng, biểu đồ lớp, các biểu đồ tương tác). Đồng thời một kiến trúc hệ thống cũng được đề xuất. Kiến trúc này có thể dựng thành nguyên mẫu, trên đó có thể thử nghiệm nhiều ý đồ với hệ thống.
- Pha xây dựng: tập trung vào việc thiết kế và thực thi (cài đặt hệ thống). Pha xây dựng kết thúc khi đã phát hành được (ít nhất trong nội bộ) một hệ thống hoàn chỉnh cùng với các tư liệu kèm theo. Pha xây dựng là pha kéo dài về thời gian và tổn hao sức lực hơn cả.
- Pha chuyển giao: Nhằm chuyển những hệ thống đã xây dựng từ tay người phát triển hệ thống tới tay người dùng cuối, bao gồm các công việc như chuyển đổi dữ liệu, đào tạo người dùng, lắp đặt, kiểm định.

Mỗi pha nói trên, đặc biệt là pha xây dựng lại được chia thành một số vòng lặp. Mỗi vòng lặp sẽ hoàn thành một phần của hệ thống và trải qua năm công đoạn sau: nắm bắt yêu cầu, phân tích và thiết kế, thực thi, kiểm định và bố trí.

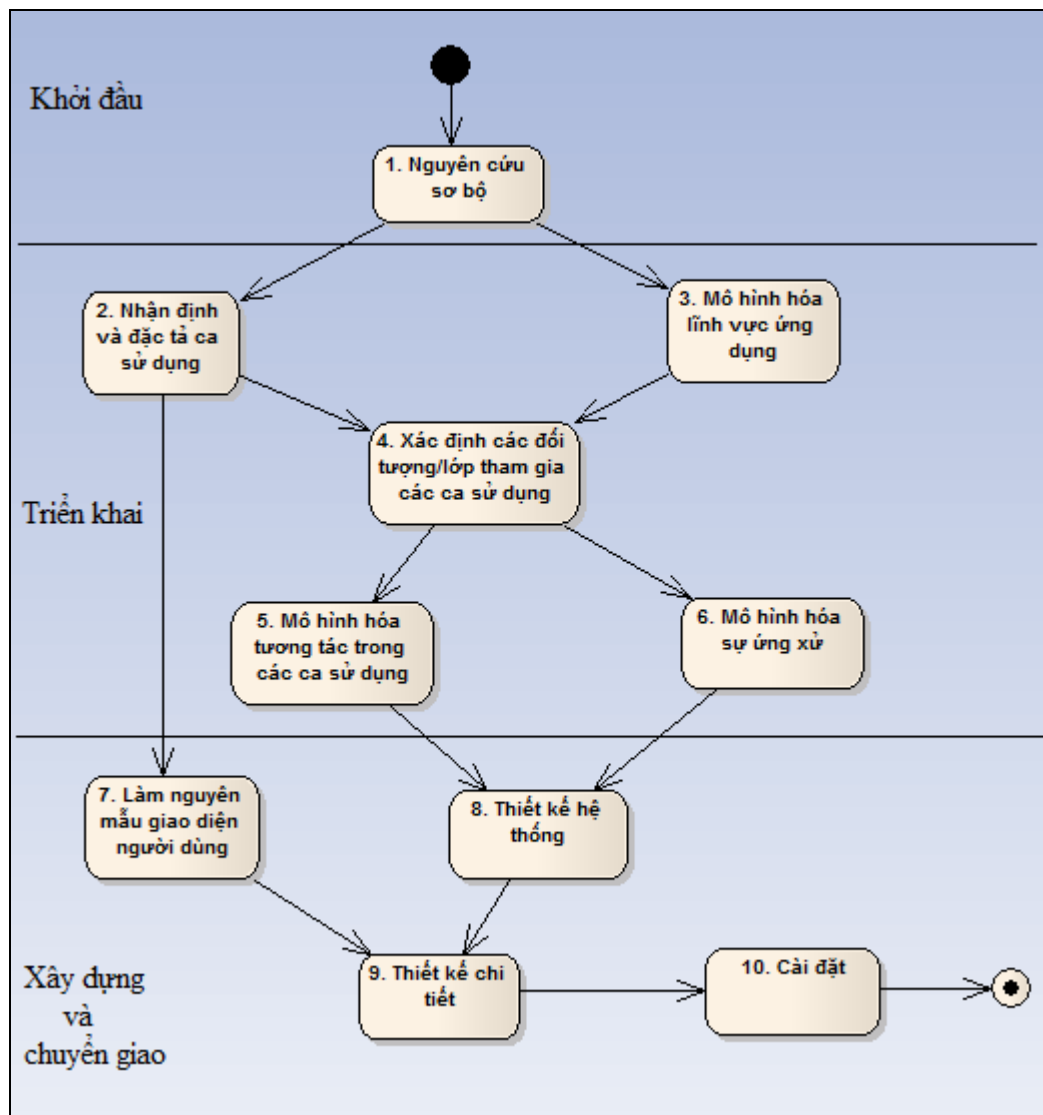
c) Một tiến trình đơn giản

Tiến trình RUP như đã nói ở trên, với các nguyên tắc cơ bản, và với sự phân chia thành các pha, các vòng lặp, các công đoạn, thì vẫn chưa phải là một tiến trình cụ thể, mà chỉ là một khuôn khổ hay một họ các tiến trình, ta còn phải chỉ rõ các bước đi, công việc phải làm trong mỗi bước, sản phẩm phải hoàn thành sau mỗi

bước, để có thể dẫn dắt một cách chắc chắn quá trình phát triển một phần mềm từ nhu cầu đặt ra cho tới mã nguồn chương trình chạy được và đáp ứng nhu cầu một cách đầy đủ và với hiệu năng mong muốn.

Sau đây là một tiến trình đơn giản, “tiến trình 10 bước”, với các đặc điểm:

- Nó thực hiện mô hình hóa với các biểu đồ UML.
- Nó tuân thủ các nguyên tắc của RUP.
- Nó chú trọng nguyên tắc dẫn dắt bởi các ca sử dụng, song cố gắng thể hiện một cách nhẹ nhàng và tự nhiên.
- Nó vẫn có thể triển khai theo nguyên tắc lặp và tăng trưởng (trong từng bước) và nguyên tắc khống chế bởi các nguy cơ.



Hình 5. Tiến trình mười bước

Sau đây là nội dung tóm tắt các bước trong tiến trình 10 bước:

- *Nghiên cứu sơ bộ*: nhằm đưa ra cái nhìn khái quát về hệ thống sẽ xây dựng (chức năng, hiệu năng, công nghệ...) và về dự án sẽ triển khai. Từ đó

đưa ra kết luận nên triển khai tiếp hay nên chấm dứt dự án. Như vậy đây chính là pha khởi đầu của RUP.

- *Nhận định và đặc tả các ca sử dụng*: Từ việc nắm bắt các nhu cầu của người dùng mà phát hiện các ca sử dụng. Ca sử dụng là một tập hợp các dãy hành động mà hệ thống thực hiện để đưa ra một kết quả có ích cho một đối tác của hệ thống. Mỗi ca sử dụng phải được đặc tả dưới dạng văn tự (kịch bản) và/hoặc dưới dạng một biểu đồ trình tự hệ thống.
- *Mô hình hóa lĩnh vực ứng dụng*: Đưa ra một mô hình (dưới dạng một biểu đồ lớp) nhằm phản ánh mọi khái niệm nghiệp vụ (thực thể và liên kết) mà người dùng cũng như người xây dựng hệ thống, khi đề cập tới hệ thống và ứng dụng đều phải đề cập đến. Các lớp xuất hiện ở đây đều là các lớp “lĩnh vực”, nghĩa là các lớp thuộc lĩnh vực nghiệp vụ của ứng dụng, mà chưa có các lớp phụ trợ khác.
- *Xác định các đối tượng /lớp tham gia các ca sử dụng*: Đối với mỗi ca sử dụng, phải phát hiện các lớp lĩnh vực, cùng các lớp điều khiển và các lớp biên tham gia thực hiện ca sử dụng đó. Như vậy ta lập một biểu đồ lớp làm nền cho mỗi ca sử dụng. Chính trên nền đó ta nghiên cứu sự tương tác ở các bước sau.
- *Mô hình hóa tương tác trong các ca sử dụng*: Sự tương tác duy nhất có thể giữa các đối tượng là trao đổi thông điệp. Cần phải nghiên cứu sự tương tác giữa các đối tượng tham gia mỗi ca sử dụng, mà kết quả là tạo nên kịch bản của ca sử dụng đó. Sự tương tác được trình bày dưới dạng biểu đồ trình tự hay biểu đồ giao tiếp.
- *Mô hình hóa ứng xử*: Các đối tượng điều khiển khác với các đối tượng thực thể ở chỗ có khả năng ứng xử trước các sự kiện từ ngoài đến đưa ra các quyết định điều khiển thích hợp. Việc mô tả hành vi ứng xử của các đối tượng điều khiển được thực hiện bởi biểu đồ máy trạng thái.
- *Làm nguyên mẫu giao diện người dùng*: Với các bộ tạo lập giao diện người dùng, ta có thể thành lập sớm và nhanh một nguyên mẫu giao diện người dùng, giúp cho việc mô hình hóa và cài đặt hệ thống triển khai dễ dàng hơn.
- *Thiết kế hệ thống*: Đó là sự thiết kế kiến trúc tổng thể của hệ thống bao gồm việc vỡ hệ thống thành các hệ thống con, chọn lựa loại hình điều khiển thích hợp, miêu tả các thành phần vật lý của hệ thống (dùng biểu đồ thành phần) và bố trí các thành phần khả thi vào phần cứng (dùng các biểu đồ bố trí). Một kiến trúc khách hàng dịch vụ nhiều tầng thường được lựa chọn ở đây.
- *Thiết kế chi tiết*: Đó là thiết kế về các lớp, các liên kết, các thuộc tính, các thao tác, thực hiện trên từng tầng của kiến trúc khách hàng/dịch vụ (tầng

khách hàng, tăng nghiệp vụ, tăng lưu trữ dữ liệu), và xác định các giải pháp cài đặt trên mạng.

- *Cài đặt:* Đó là bước thực thi hệ thống, bao gồm lập trình và kiểm định. Hệ thống được nghiệm thu dựa trên các ca sử dụng.

PHẦN II. CÁC KẾT QUẢ ĐẠT ĐƯỢC

CHƯƠNG 3. KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

3.1 Các yêu cầu của hệ thống

Phần mềm lập lịch là một công cụ hướng tới người dùng chủ yếu là các doanh nghiệp, nhóm người cùng làm việc hợp tác với nhau, giúp họ tạo các lịch làm việc cá nhân, lịch làm việc liên quan tới nhiều người, lịch cơ quan, lịch trực, ... giúp cho quá trình quản lý lịch làm việc trong tổ chức được thuận lợi và đem lại hiệu quả cao trong công việc.

Yêu cầu đặt ra đối với hệ thống lập lịch trước hết là phải đơn giản, dễ sử dụng và phù hợp với mọi cá nhân người dùng ở mọi trình độ tin học, đồng thời phải đáp ứng được đầy đủ các yêu cầu về chức năng trong phạm vi phục vụ nhu cầu tạo lịch làm việc cho cá nhân, tập thể.

3.1.1 Các yêu cầu chức năng

Qua khảo sát người dùng, phần mềm cần đảm bảo các chức năng sau:

➤ Quản lý lịch làm việc

- Có thể tạo lịch cá nhân, lịch làm việc cho người khác, lịch cho một nhóm người một cách nhanh chóng.
- Khi tạo lịch có gợi ý về khoảng thời gian trống của một người, một nhóm người, tài nguyên để giúp cho lịch không bị xung đột.
- Khi lịch có xung đột (xung đột về người tham gia, về tài nguyên...) thì có thông báo cho người tạo lịch.
- Có thể sửa lịch một cách dễ dàng: thêm bớt người tham gia, nhóm tham gia, tài nguyên.
- Có khả năng nhắc lịch, thời gian nhắc này có thể tùy biến được.
- Có thể tạo được các lịch có tính chất định kỳ (lặp lại theo ngày, tháng...).
- Có thể tạo lịch riêng tư (với những lịch này chỉ người tham gia mới xem được nội dung), lịch công cộng (mọi người trong hệ thống đều có thể xem được thông tin lịch. Mặc định là lịch công cộng).
- Người tham gia lịch có thể đồng ý từ chối lịch do người khác lập. Lịch của mình lập ra thì mặc định là đồng ý.
- Trong lịch có thể đính kèm tài liệu dưới dạng tệp tin chia sẻ (tệp tin này có vai trò như là tài liệu chuẩn bị trước cho cuộc họp).
- Có thể tìm kiếm lịch theo người tạo, theo thời gian, theo thành phần tham gia...

- Có thể lọc được lịch dựa vào một số tiêu chí: thời gian diễn ra, thành phần tham gia, lịch đã đồng ý, lịch đã từ chối, lịch chưa trả lời... giúp cho việc xem thông tin lịch dễ dàng.
- **Thông báo lịch**
 - Khi tạo lịch, thay đổi lịch thì có thông báo đến những người liên quan.
 - Hỗ trợ nhiều các thông báo lịch: thông báo qua notify trong hệ thống, qua email, qua tin nhắn SMS...
- **Quản lý tài nguyên**
 - Cho phép thêm, xóa, sửa các loại tài nguyên của cơ quan, chọn người được phép sử dụng, người duyệt tài nguyên này.
 - Khi có một lịch liên quan đến tài nguyên thì người duyệt tài nguyên sẽ được thông báo.
- **Đồng bộ**
 - Mọi người trong cơ quan cùng làm việc với nhau nên thông tin thống nhất là cần thiết. Để giải quyết vấn đề đó, phần mềm được tổ chức dạng client-server, lưu trữ dữ liệu tập trung ở server và đồng bộ khi người dùng đăng nhập vào hệ thống.
 - Khi người dùng đăng nhập vào hệ thống, mọi dữ liệu sẽ được đồng bộ giữa client và server (lấy những thay đổi từ sever về client. Sau đó upload những thay đổi từ client lên server).
- **Thống kê**
 - Phần mềm hỗ trợ người dùng thống kê các lịch làm việc trong ngày, tuần, tháng...
- **Cập nhật**
 - Phần mềm có thể cập nhật lên phiên bản mới mà không cần tải về toàn bộ và cài đặt mới.
- **Quản lý**
 - Người dùng có nhiều tùy chọn để quản lý các thành phần của hệ thống theo ý muốn. Bao gồm:
 - Quản lý tài khoản: thay đổi thông tin người dùng như thay đổi mật khẩu.
 - Có các lựa chọn để thay đổi màu sắc giao diện.
 - Thay đổi số ngày hiển thị lịch trong một tuần.

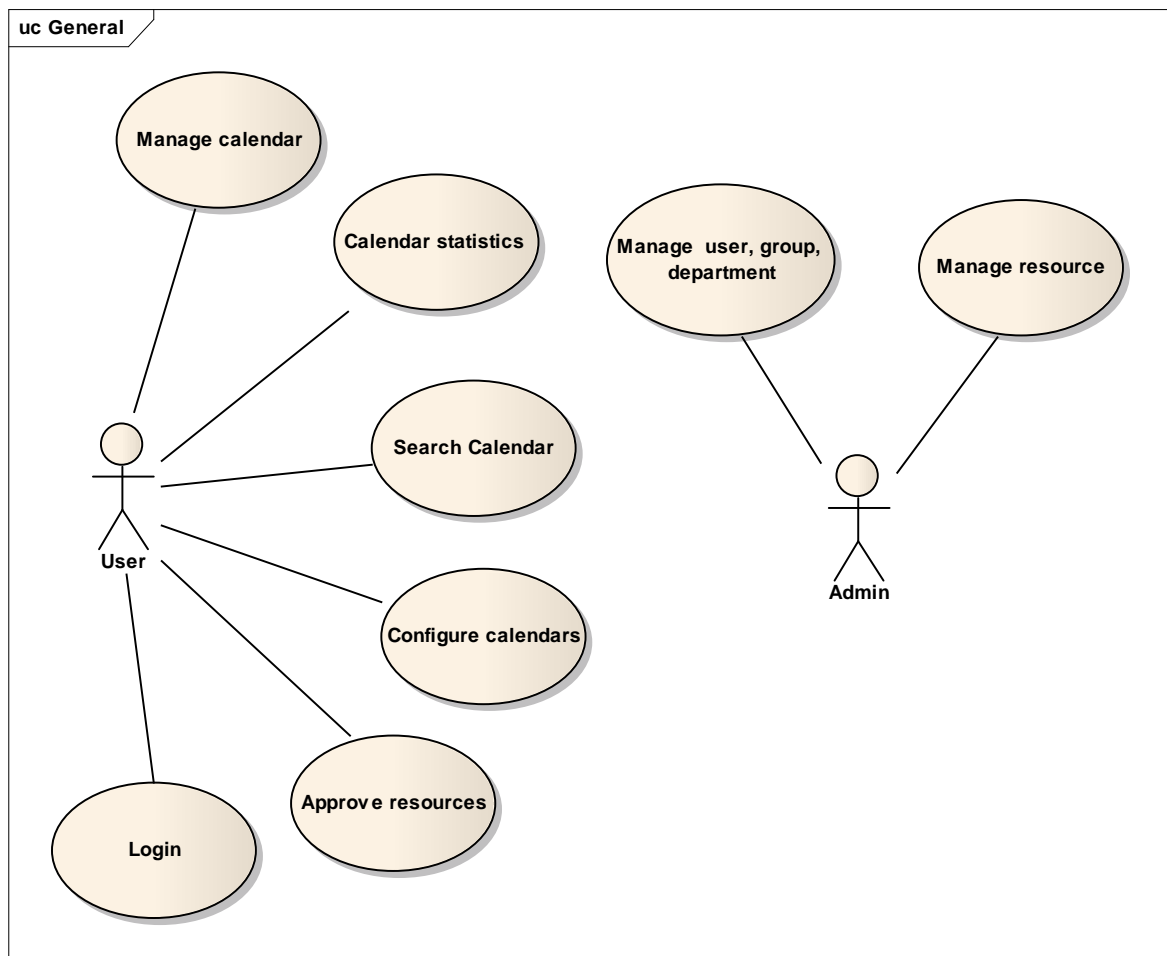
3.1.2 Các yêu cầu phi chức năng

Ngoài các yêu cầu về chức năng như trên hệ thống có các yêu cầu phi chức năng như:

- Hệ thống gọn nhẹ không tốn tài nguyên.
- Hỗ trợ tiếng Việt và bộ gõ tiếng Việt trong chương trình để phù hợp với người dùng ở mọi trình độ.
- Giao diện thân thiện, bố trí phù hợp.
- Việc hiển thị lịch: phân loại lịch theo thời gian (như ngày, tuần, tháng...) giúp cho việc theo dõi, xem thông tin lịch một cách nhanh chóng, thuận tiện.

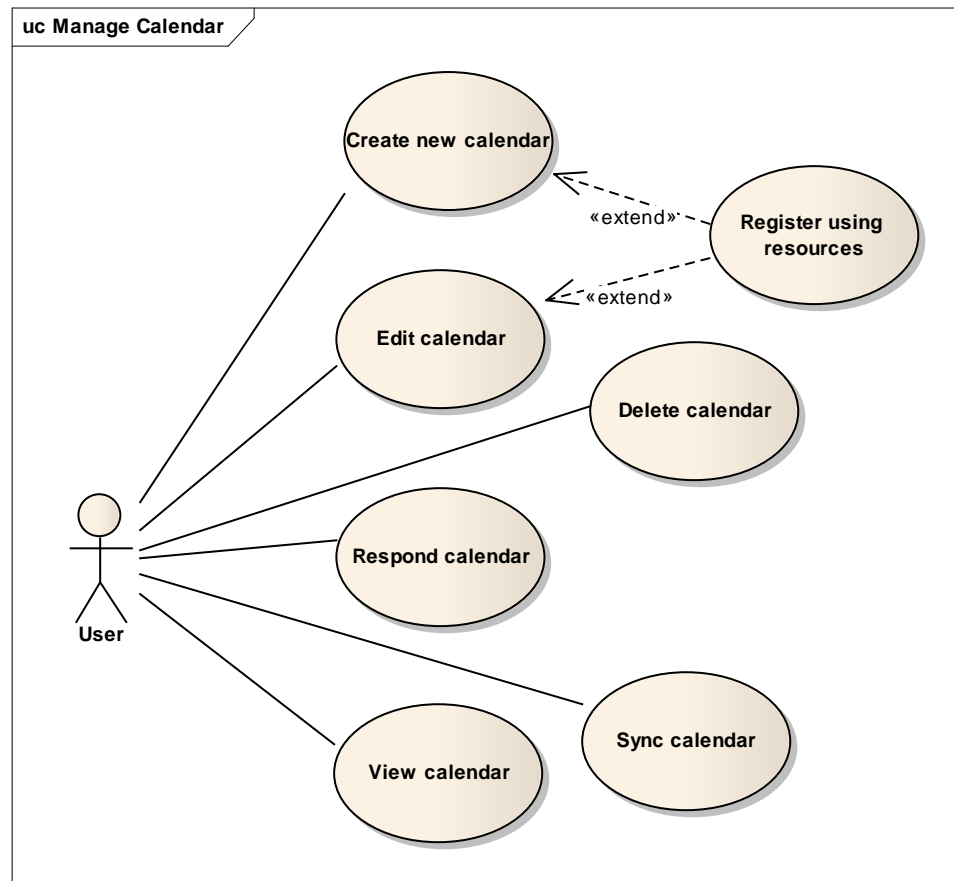
3.2 Nhận định và đặc tả các ca sử dụng

3.2.1 Biểu đồ ca sử dụng tổng quan



Hình 6. Biểu đồ ca sử dụng tổng quan

3.2.2 Biểu đồ ca sử dụng quản lý lịch



Hình 7. Biểu đồ ca sử dụng quản lý lịch

Mô tả tóm tắt

- Tên ca sử dụng: Quản lý lịch.
- Mục đích: Giúp người dùng hệ thống tạo lịch mới, sửa lịch, xóa lịch, xem lịch, trả lời lịch.
- Tóm lược: Người dùng có thể tạo lịch mới, sửa lịch, xóa lịch, xem lịch, trả lời lịch.
- Đối tác: User.

Mô tả các kịch bản

- *Điều kiện đầu vào*

Ca sử dụng này, chỉ có thể thực hiện khi người dùng đã đăng nhập hệ thống.

- *Kịch bản chính*

Ca sử dụng bắt đầu khi người dùng đăng nhập phần mềm (R2.1). Tại giao diện chính người dùng có thể thực hiện các chức năng quản lý lịch.

Nếu người dùng nhấp đúp chuột vào một lịch thì kịch bản con C2.1: Xem lịch được thực hiện.

Nếu người dùng tạo lịch lịch mới thì kịch bản con C2.2: Tạo một lịch mới được thực hiện.

Nếu người dùng chọn sửa lịch thì kịch bản con C2.3: Sửa lịch được thực hiện.

Nếu người dùng chọn xóa lịch thì kịch bản con C2.4: Xóa lịch được thực hiện.

Nếu người dùng trả lời lịch thì kịch bản con C2.5: Trả lời lịch được thực hiện.

Nếu người dùng chọn đồng bộ lịch thì kịch bản con C2.6: Trả lời lịch được thực hiện.

- *Kịch bản con*

C2.1: Xem lịch

Nếu người dùng là người tạo lịch thì cửa sổ chi tiết lịch (có thể chỉnh sửa thông tin) hiện lên với đầy đủ các thông tin về lịch. Nếu người dùng là người tham gia thì cửa sổ chi tiết lịch (chỉ có thể xem thông tin) hiện lên với đầy đủ thông tin của lịch. Người dùng nhấn “thoát” để đóng cửa sổ. Ca sử dụng bắt đầu lại.

C2.2: Tạo một lịch mới

Hệ thống hiển thị cửa sổ tạo lịch mới bao gồm các thông tin cần thiết cho một lịch như: tiêu đề lịch, thời gian bắt đầu, thời gian kết thúc, địa điểm, thành phần tham gia (tài nguyên, người tham gia), tùy chọn cho lịch (nhắc lịch, lịch riêng tư, lịch lặp...), ...Người dùng điền các thông tin này và nhấn nút “tạo lịch” để tạo một lịch mới (R2.2) hoặc “bỏ qua” để thoát. Ca sử dụng bắt đầu lại.

C2.3: Sửa lịch

Hệ thống hiển thị cửa sổ chi tiết lịch (có thể chỉnh sửa thông tin) hiện lên, bao gồm đầy đủ thông tin về lịch như: tiêu đề lịch, thời gian bắt đầu, thời gian kết thúc, địa điểm, thành phần tham gia, tùy chọn cho lịch (nhắc lịch, lịch riêng tư, lịch lặp...), ...Người dùng sửa các thông tin này và nhấn nút “cập nhật” để lưu thay đổi (R2.2) hoặc “bỏ qua” để thoát. Ca sử dụng bắt đầu lại.

C2.4: Xóa lịch

Nếu có quyền xóa lịch thì hệ thống hiển lên thông báo xác nhận xóa lịch, người dùng chọn “đồng ý” để xóa lịch khỏi hệ thống, “bỏ qua” để không xóa lịch. Ca sử dụng bắt đầu lại.

C2.5: Trả lời lịch

Người dùng là người tham gia lịch thì có quyền trả lời. Người dùng có các lựa chọn trả lời: đồng ý, từ chối. Khi người dùng trả lời lịch hệ thống sẽ cập nhật lại trạng thái của lịch. Ca sử dụng bắt đầu lại.

C2.6: Đồng bộ lịch

Với những lịch chưa được đồng bộ, lịch sẽ được gửi lên server để đồng bộ.

- *Kịch bản khả dĩ khác*

R2.1:

Mật khẩu hoặc tên đăng nhập của người dùng không đúng, người dùng phải điền lại mật khẩu, tên tài khoản hoặc kết thúc ca sử dụng.

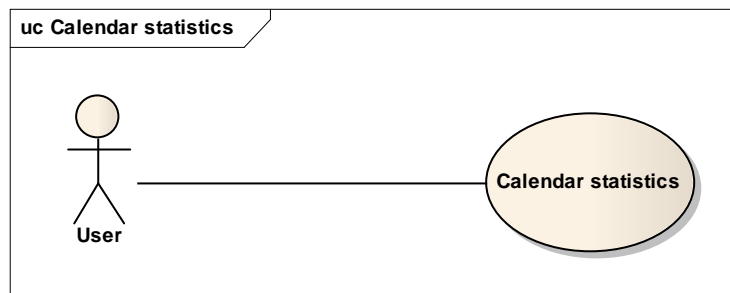
Hoặc không kết nối được tới server, người dùng phải đăng nhập lại.

R2.2:

Thông tin của lịch không đúng chuẩn hoặc lịch bị xung đột thì hệ thống báo cho người dùng, người dùng sẽ thay đổi thông tin của lịch để loại trừ xung đột hoặc phớt lờ xung đột đó.

Lỗi kết nối tới server, lịch sẽ được lưu tại client.

3.2.3 Biểu đồ ca sử dụng thống kê lịch



Hình 8. Biểu đồ ca sử dụng thống kê lịch

Mô tả tóm tắt

- Tên ca sử dụng: Thống kê lịch.
- Mục đích: Thống kê các lịch làm việc theo khoảng thời gian, lịch sắp phải thực hiện.
- Đối tác: User.

Mô tả các kịch bản

- *Điều kiện các đầu vào*

Ca sử dụng này, chỉ có thể thực hiện khi người dùng đã đăng nhập hệ thống.

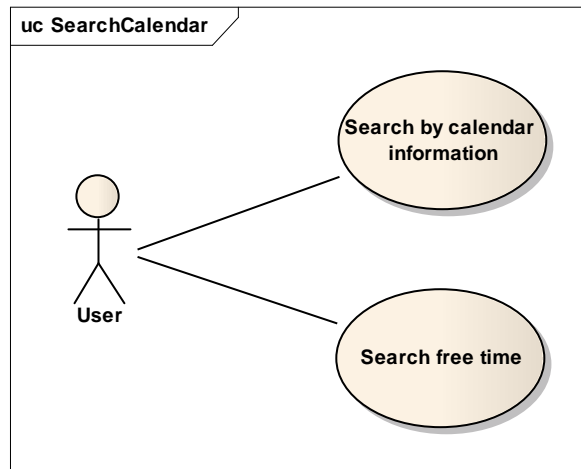
- *Kịch bản chính*

Ca sử dụng bắt đầu khi người dùng đăng nhập phần mềm (R2.1). Tại giao diện chính người dùng chọn chức năng thống kê lịch. Trên màn hình thống kê lịch người dùng có thể chọn khoảng thời gian diễn ra của lịch, trạng thái của lịch,...Nhấn nút “thống kê”, hệ thống sẽ liệt kê các lịch theo tiêu chí mà người dùng đã chọn (R3.1).

- *Kịch bản con*
- *Kịch bản khả dĩ khác*

R3.1: lỗi không kết nối được với server, hệ thống thông báo với người dùng.
Ca sử dụng kết thúc.

3.2.4 Biểu đồ ca sử dụng tìm kiếm lịch



Hình 9. Biểu đồ ca sử dụng tìm kiếm lịch

Mô tả tóm tắt

- Tên ca sử dụng: Tìm kiếm.
- Mục đích: Tìm kiếm lịch làm việc theo một tiêu chí nào đó hoặc tìm kiếm thời gian trống của một người, nhóm người, tài nguyên nào đó phục vụ cho việc đặt lịch.
- Đối tác: User.

Mô tả các kịch bản

- *Điều kiện các đầu vào*

Ca sử dụng này, chỉ có thể thực hiện khi người dùng đã đăng nhập hệ thống.

- *Kịch bản chính*

Ca sử dụng bắt đầu khi người dùng đăng nhập phần mềm (R2.1). Tại giao diện chính người dùng có thể chọn chức năng tìm kiếm lịch. Nếu người dùng chọn tìm kiếm lịch thì kịch bản con C4.1: Tìm kiếm lịch được thực hiện. Trên cửa sổ tạo lịch mới nếu người dùng chọn chức năng tìm thời gian trống thì kịch bản con C4.2: Tìm thời gian trống được thực hiện.

- *Kịch bản con*

C4.1: Tìm kiếm lịch

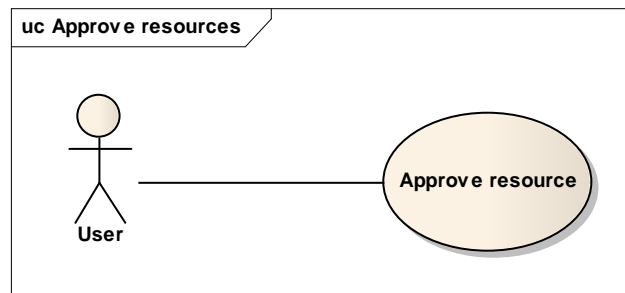
Hệ thống hiển thị màn hình tìm kiếm với những thông tin liên quan tới lịch làm việc như tiêu đề, khoảng thời gian diễn ra, thành phần tham gia, ... người dùng điền các thông tin này, nhấn nút tìm kiếm để tiến hành tìm kiếm lịch. Hệ thống sẽ hiển thị danh sách lịch theo các tiêu chí người dùng nhập vào.

C4.2: Tìm thời gian trống

Trên cửa sổ tạo lịch mới, hoặc trên giao diện chính người dùng chọn chức năng tìm thời gian trống. Hệ thống hiển thị cửa sổ tìm thời gian trống với những thông tin cần nhập như sau: tên người, tên tài nguyên, thời gian diễn ra lịch, khoảng thời gian. Người dùng nhập những thông tin này, nhấn nút tìm thời gian trống. Hệ thống sẽ hiển thị danh sách kết quả là các khoảng thời gian mà khi đó người, tài nguyên rảnh.

- *Kịch bản khả dĩ khác*

3.2.5 Biểu đồ ca sử dụng duyệt tài nguyên



Hình 10. Biểu đồ ca sử dụng duyệt tài nguyên

Mô tả tóm tắt

- Tên ca sử dụng: Duyệt tài nguyên.
- Mục đích: Phê duyệt tài nguyên cho lịch.
- Đối tác: User (người dùng được phân công duyệt tài nguyên).

Mô tả các kịch bản

- *Điều kiện các đầu vào*

Ca sử dụng này, chỉ có thể thực hiện khi người dùng đã đăng nhập hệ thống và người dùng được giao quyền duyệt một tài nguyên nào đó.

- *Kịch bản chính*

Ca sử dụng bắt đầu khi người dùng đăng nhập phần mềm (R2.1). Hệ thống liệt kê danh sách lịch cần duyệt trên giao diện duyệt tài nguyên. Nếu người dùng chọn đồng ý thì kịch bản con C5.1: Đồng ý tài nguyên được thực hiện, nếu người dùng chọn từ chối thì kịch bản con C5.2: Từ chối tài nguyên được thực hiện. Khi từ chối một lịch thì phải đưa ra lý do.

- *Kịch bản con*

C5.1: Đồng ý tài nguyên của lịch

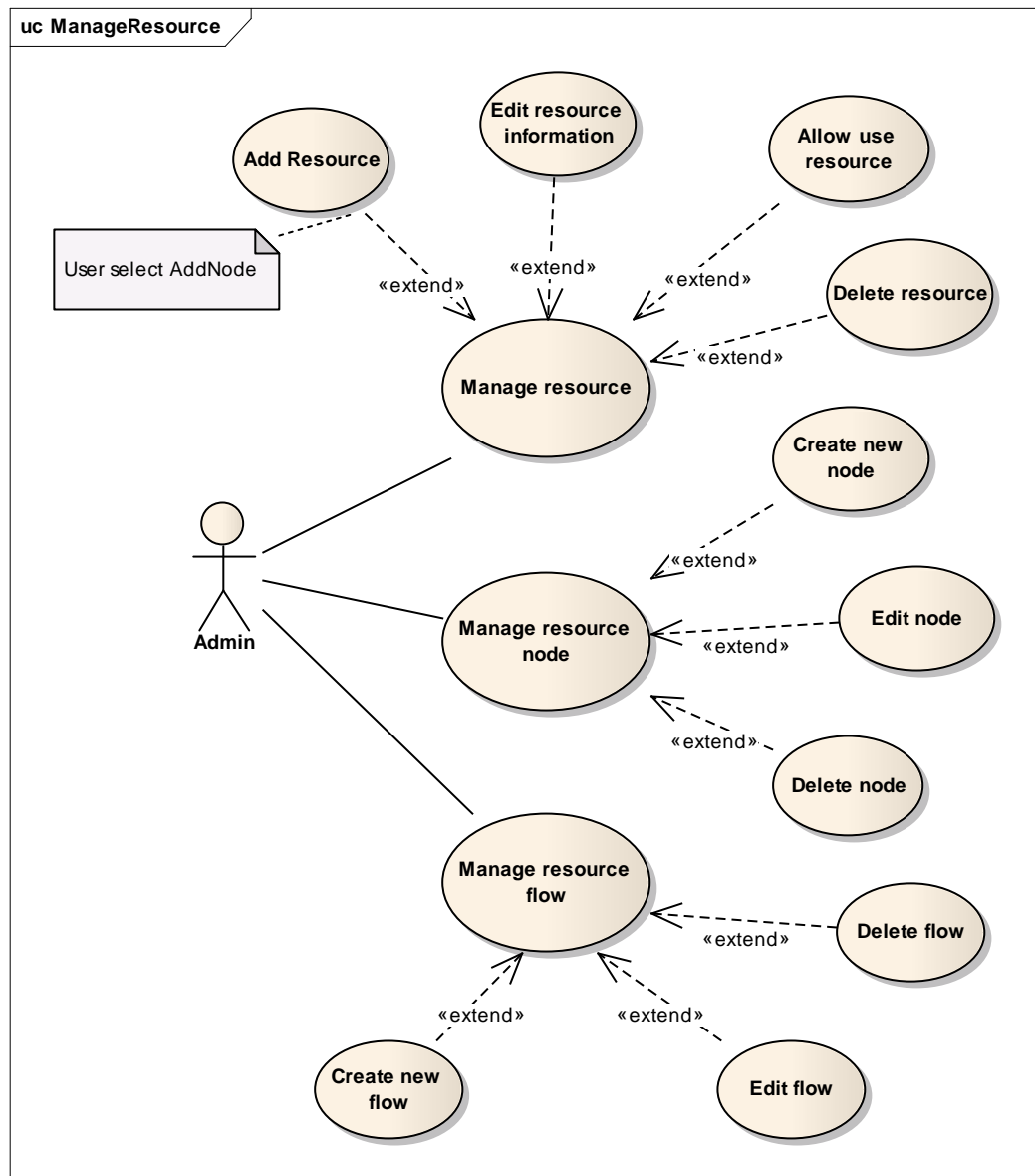
Hệ thống sẽ cập nhật lại trạng thái của lịch.

C5.2: Từ chối tài nguyên

Hệ thống yêu cầu người dùng nhập lý do từ chối. Người dùng nhập lý do và bấm cập nhật để từ chối tài nguyên.

- *Kịch bản khả dĩ khác*

3.2.6 Biểu đồ ca sử dụng quản lý tài nguyên



Hình 11. Biểu đồ ca sử dụng quản lý tài nguyên

Mô tả tóm tắt

- Tên ca sử dụng: Quản lý tài nguyên.
- Mục đích: Quản lý thông tin tài nguyên của cơ quan.
- Đối tác: Admin (người quản trị hệ thống).

Mô tả các kịch bản

- *Điều kiện các đầu vào*

Ca sử dụng này chỉ có thể bắt đầu khi người quản trị đăng nhập vào server.

- *Kịch bản chính*

Người quản trị đăng nhập vào server. Tại giao diện chính quản trị viên chọn chức năng quản lý tài nguyên. Hệ thống hiển thị cửa sổ quản lý tài nguyên: trên đó gồm có danh sách tài nguyên, danh sách luồng xét duyệt,

node xét duyệt, ... với các thông tin chi tiết về tài nguyên, luồng xét duyệt, node xét duyệt.

Nếu người dùng chọn thêm tài nguyên thì kịch bản con C7.1: Thêm tài nguyên được thực hiện.

Nếu người dùng chọn xóa tài nguyên thì kịch bản con C7.2: Xóa tài nguyên được thực hiện.

Nếu người dùng chọn sửa thông tin tài nguyên thì kịch bản con C7.3: Sửa thông tin tài nguyên được thực hiện.

Nếu người dùng chọn quản lý node xét duyệt thì kịch bản con C7.4: Quản lý node xét duyệt được thực hiện.

Nếu người dùng chọn quản lý luồng xét duyệt thì kịch bản con C7.5: Quản lý luồng xét duyệt được thực hiện.

- *Kịch bản con*

C7.1: Thêm tài nguyên được thực hiện

Hệ thống hiển thị cửa sổ thêm tài nguyên với các thông tin: tên tài nguyên, node liên quan tới tài nguyên, luồng xét duyệt tài nguyên, node người sử dụng tài nguyên, loại tài nguyên... Người dùng nhập những thông tin này, bấm cập nhật để thêm tài nguyên. Tài nguyên sẽ được thêm vào hệ thống.

C7.2: Xóa tài nguyên được thực hiện

Hệ thống hiển thị danh sách tài nguyên, người dùng chọn một tài nguyên bấm xóa để xóa tài nguyên, tài nguyên sẽ bị xóa khỏi hệ thống.

C7.3: Sửa thông tin tài nguyên được thực hiện

Hệ thống hiển thị chi tiết thông tin của tài nguyên với các thông tin: tên tài nguyên, node liên quan tới tài nguyên, luồng xét duyệt tài nguyên, node người sử dụng tài nguyên, loại tài nguyên... Người dùng chỉnh sửa những thông tin này, bấm cập nhật để sửa tài nguyên.

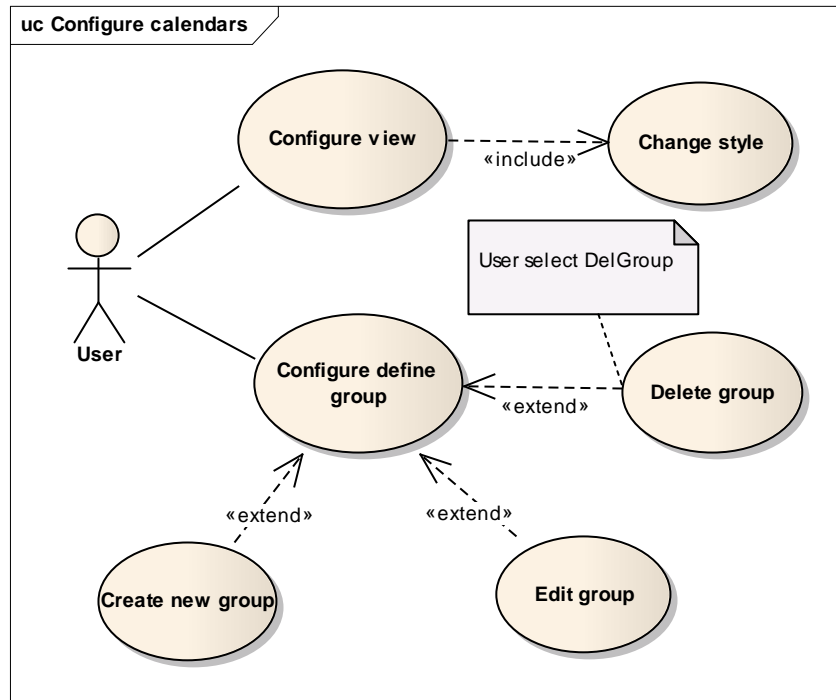
C7.4: Quản lý node xét duyệt được thực hiện

Hệ thống hiển thị cửa sổ quản lý node xét duyệt tài nguyên, trên đó hiển thị danh sách các node xét duyệt. Người dùng chọn vào node nào thì thông tin về node đó sẽ được hiển thị, những thông tin về node xét duyệt đó như: tên node, danh sách người xét duyệt tại node, chức vụ xét duyệt tại node. Người dùng có thể thêm, sửa xóa node.

C7.5: Quản lý luồng xét duyệt được thực hiện

Hệ thống hiển thị cửa sổ quản lý luồng xét duyệt tài nguyên, trên đó hiển thị danh sách các luồng xét duyệt. Người dùng chọn vào luồng nào thì thông tin chi tiết về luồng đó sẽ được hiển thị, những thông tin của một luồng xét duyệt đó như: tên luồng, danh sách các node trên luồng. Người dùng có thể thêm, sửa xóa node.

3.2.7 Biểu đồ ca sử dụng cấu hình phân lập lịch



Hình 12. Biểu đồ ca sử dụng cấu hình phân lập lịch

Mô tả tóm tắt

- Tên ca sử dụng: Cấu hình phân lập lịch.
- Mục đích: Tùy chọn các cách hiển thị khác nhau, quản lý nhóm tự định nghĩa (các nhóm tự định nghĩa là nhóm các người dùng, tài nguyên, nhóm này giúp cho việc tạo lịch làm việc nhanh hơn).
- Đối tác: User.

Mô tả các kịch bản

- *Điều kiện các đầu vào*

Ca sử dụng này, chỉ có thể thực hiện khi người dùng đã đăng nhập phần mềm.

- *Kịch bản chính*

Ca sử dụng bắt đầu khi người dùng đăng nhập phần mềm (R2.1). Từ giao diện chính, nếu người dùng chọn chức năng tùy chọn lịch thì kịch bản con C7.1: Tùy chọn lịch được thực hiện, nếu người dùng chọn quản lý nhóm tự định nghĩa thì kịch bản con C7.2: Quản lý nhóm tự định nghĩa được thực hiện.

- *Kịch bản con*

C7.1: Tùy chọn lịch

Hệ thống hiển thị cửa sổ với các tùy chọn về giao diện lập lịch như màu sắc, số ngày hiển thị lịch trong tuần, thời gian bắt đầu và kết thúc ngày làm việc...Người dùng có thể thay đổi các tùy chọn này cho phù hợp với bản

thân, bấm “lưu” để lưu các thông tin thay đổi, các thông tin này sẽ được sử dụng cho các lần đăng nhập sau.

C7.2: Quản lý nhóm tự định nghĩa

Hệ thống hiển thị màn hình quản lý các nhóm tự định nghĩa trên đó có danh sách các nhóm mà người dùng đã định nghĩa, người dùng chọn vào mỗi nhóm để xem thông tin chi tiết của nhóm. Thông tin chi tiết của nhóm bao gồm: danh sách người dùng trong nhóm, tài nguyên trong nhóm. Người dùng có thể thêm, sửa xóa các nhóm tự định nghĩa này. Thông tin các nhóm sẽ được đồng bộ với server.

- *Kịch bản khả dĩ khác*

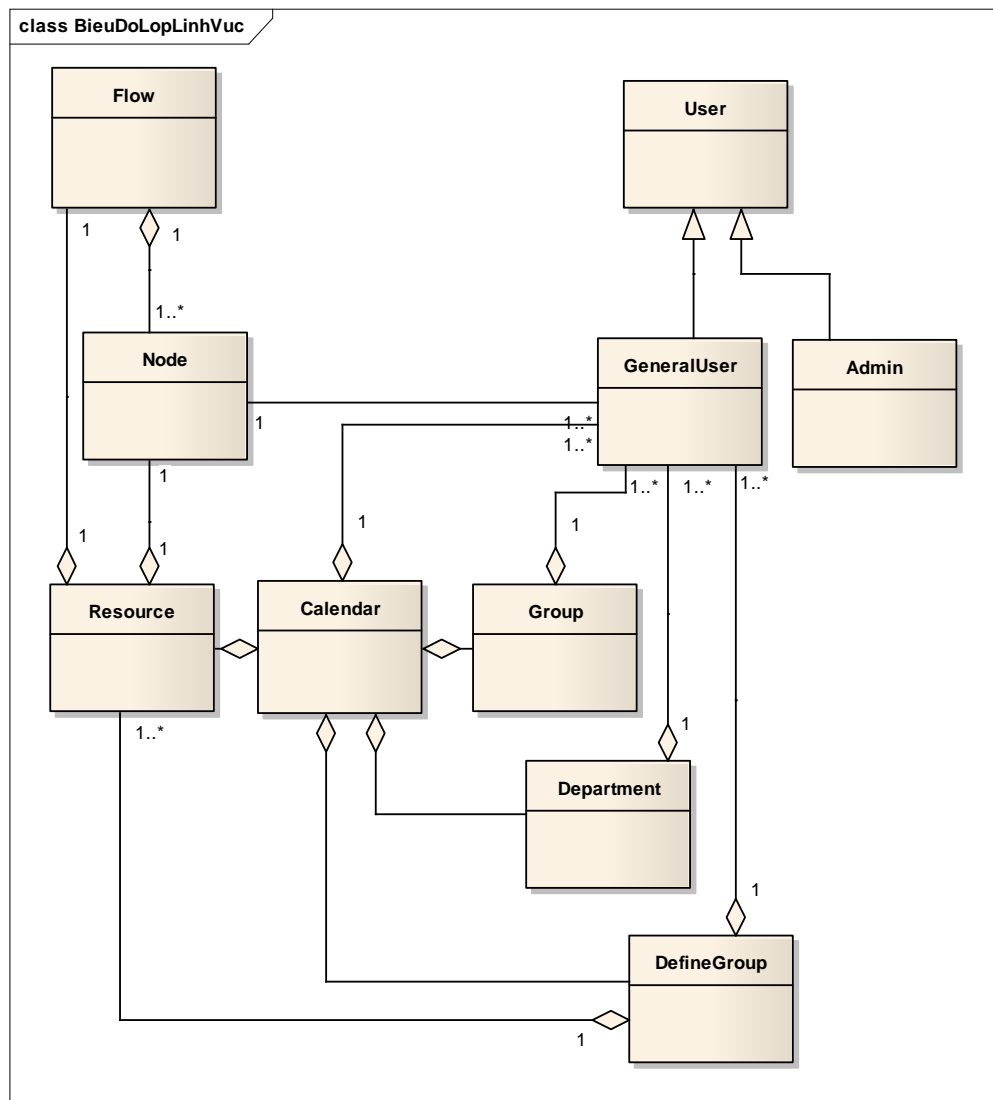
3.3 Mô hình hóa cấu trúc

Là diễn tả khía cạnh tĩnh của hệ thống. Trong phần này ta chỉ xét đến sự phát hiện và khẳng định vai trò của các lớp một cách sơ bộ.

3.3.1 Mô hình hóa lĩnh vực ứng dụng

Mục đích của mô hình hóa ứng dụng là thực hiện mô hình hóa từ vựng của hệ thống. Xuất phát từ các khái niệm về sự vật trong lĩnh vực ứng dụng, ta trừu tượng hóa chúng thành các lớp gọi là các lớp lĩnh vực. Các lớp này dùng để phản ánh và mô phỏng các sự vật trong thế giới thực.

Hệ thống lập lịch làm việc có các lớp lĩnh vực như hình vẽ sau:



Hình 13. Biểu đồ lớp lĩnh vực

3.3.2 Xác định các đối tượng/lớp tham gia các ca sử dụng

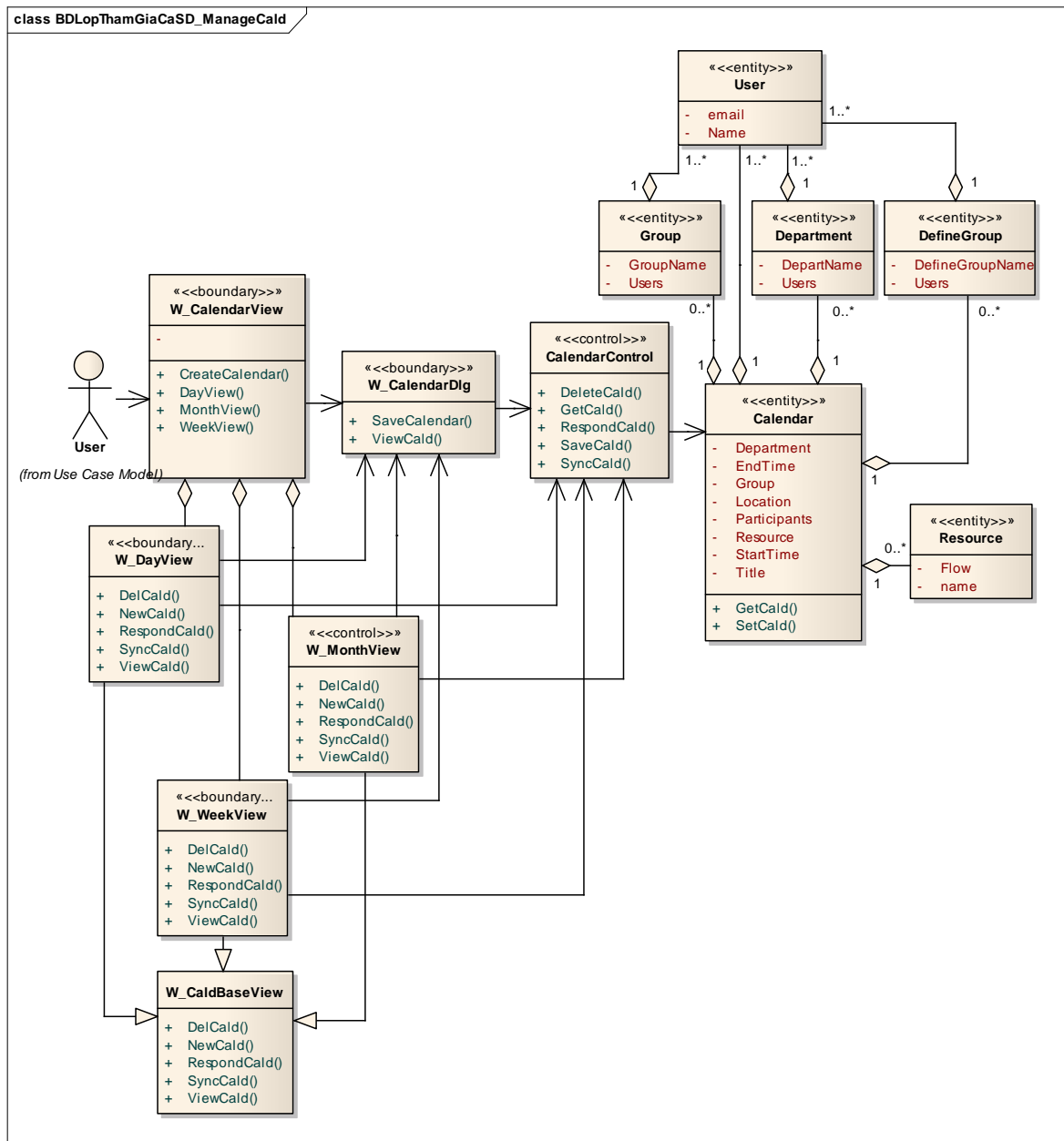
Mục đích của phần này là giúp ta khẳng định được vị thế của các lớp lĩnh vực ứng dụng trong hệ thống, đồng thời phát hiện thêm các loại đối tượng phụ trợ, như các đối tượng biên và các đối tượng điều khiển.

a) Biểu đồ lớp tham gia của ca sử dụng ‘Quản lý lịch’

Từ kịch bản của ca sử dụng ta phát hiện các lớp tham gia như sau:

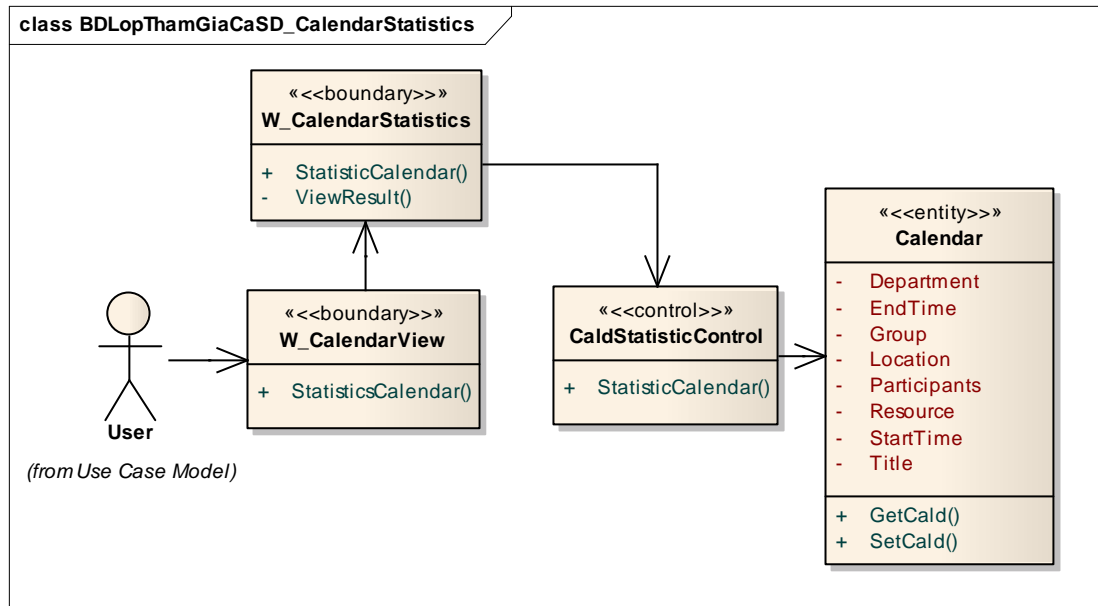
- Các lớp thực thể, lấy từ biểu đồ lĩnh vực sang và có mặt ở đây bởi các thông tin của nó được đề cập trong kịch bản, là: User, Calendar, Resource, Group, Department, DefineGroup.
- Các lớp biên gồm:
 - Lớp W_CalendarView là màn hình chính của phần mềm.
 - Lớp W_DayView, W_WeekView, W_MonthView là các màn hình hiển thị lịch theo nhóm các lịch (lịch trong ngày, trong tuần, trong tháng).

- W_CalendarDlg cửa sổ tạo một lịch mới.
- Lớp điều khiển cần có một, lấy tên là CalendarControl.



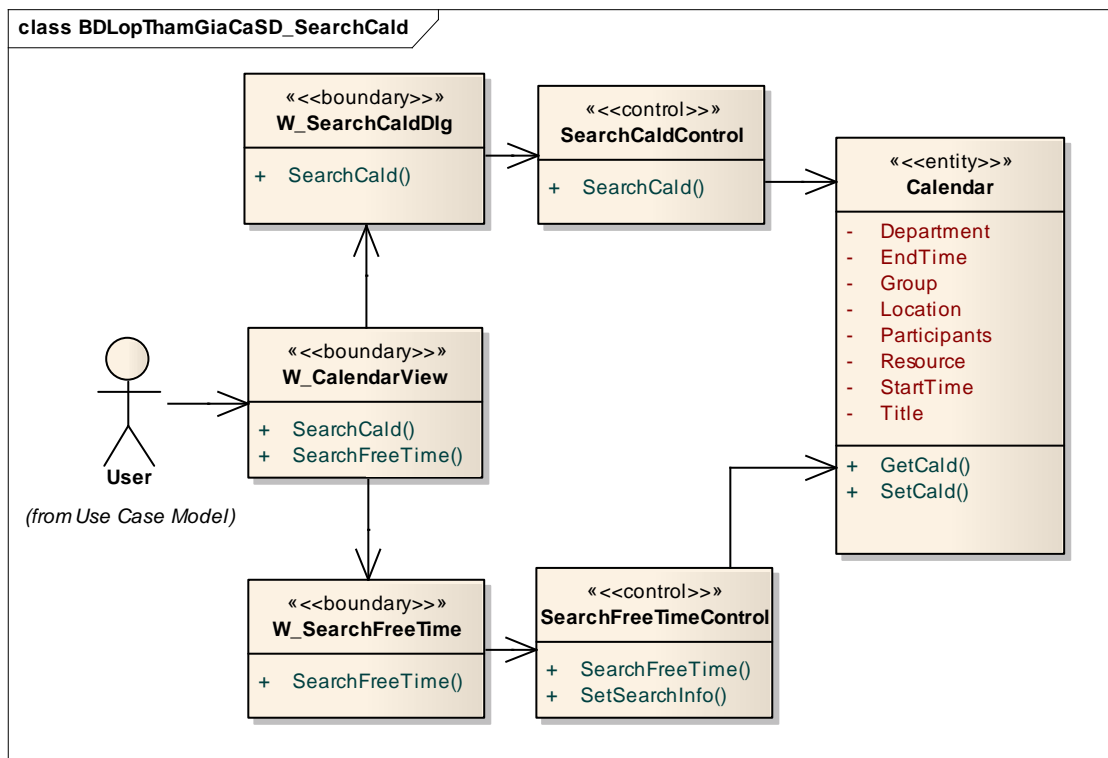
Hình 14. Biểu đồ lớp tham gia của ca sử dụng 'Quản lý lịch'

b) Biểu đồ lớp tham gia của ca sử dụng 'Thống kê lịch'



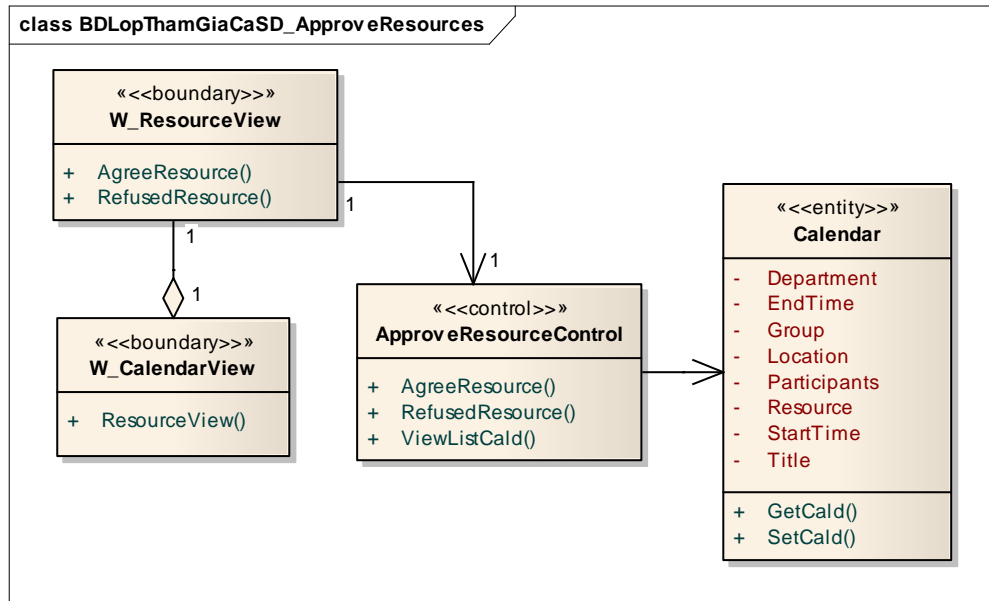
Hình 15. Biểu đồ lớp tham gia của ca sử dụng 'Thống kê lịch'

c) Biểu đồ lớp tham gia của ca sử dụng 'Tìm kiếm lịch'



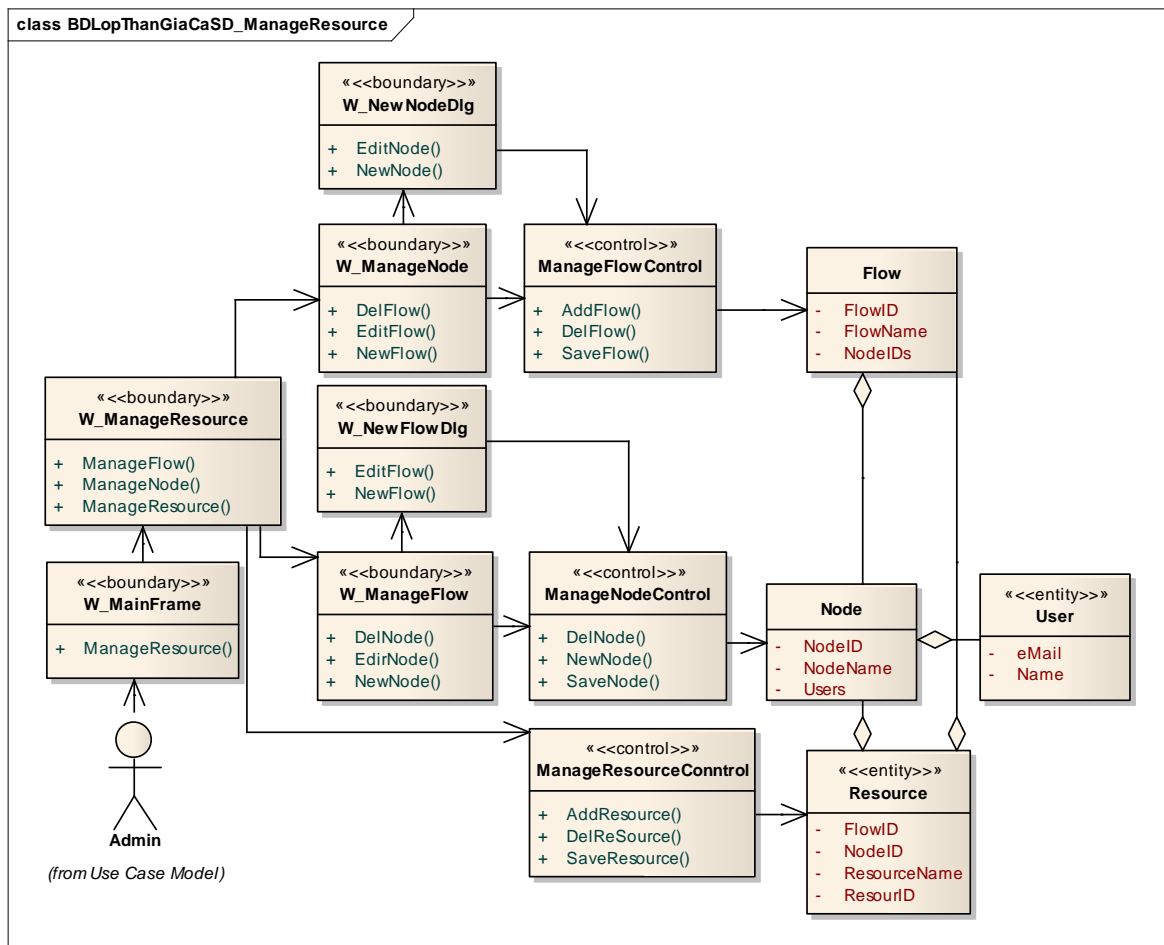
Hình 16. Biểu đồ lớp tham gia của ca sử dụng 'Tìm kiếm lịch'

d) Biểu đồ lớp tham gia của ca sử dụng 'Duyệt tài nguyên'



Hình 17. Biểu đồ lớp tham gia của ca sử dụng 'Duyệt tài nguyên'

e) Biểu đồ lớp tham gia của ca sử dụng 'Quản lý tài nguyên'



Hình 18. Biểu đồ lớp tham gia của ca sử dụng 'Quản lý tài nguyên'

3.4 Mô hình hóa hành vi

Trong phần này ta sẽ diễn tả khía cạnh động của hệ thống, đối lập với mô hình hóa cấu trúc, là sự diễn tả khía cạnh tĩnh của hệ thống.

Có ba cách tiếp cận hành vi của hệ thống:

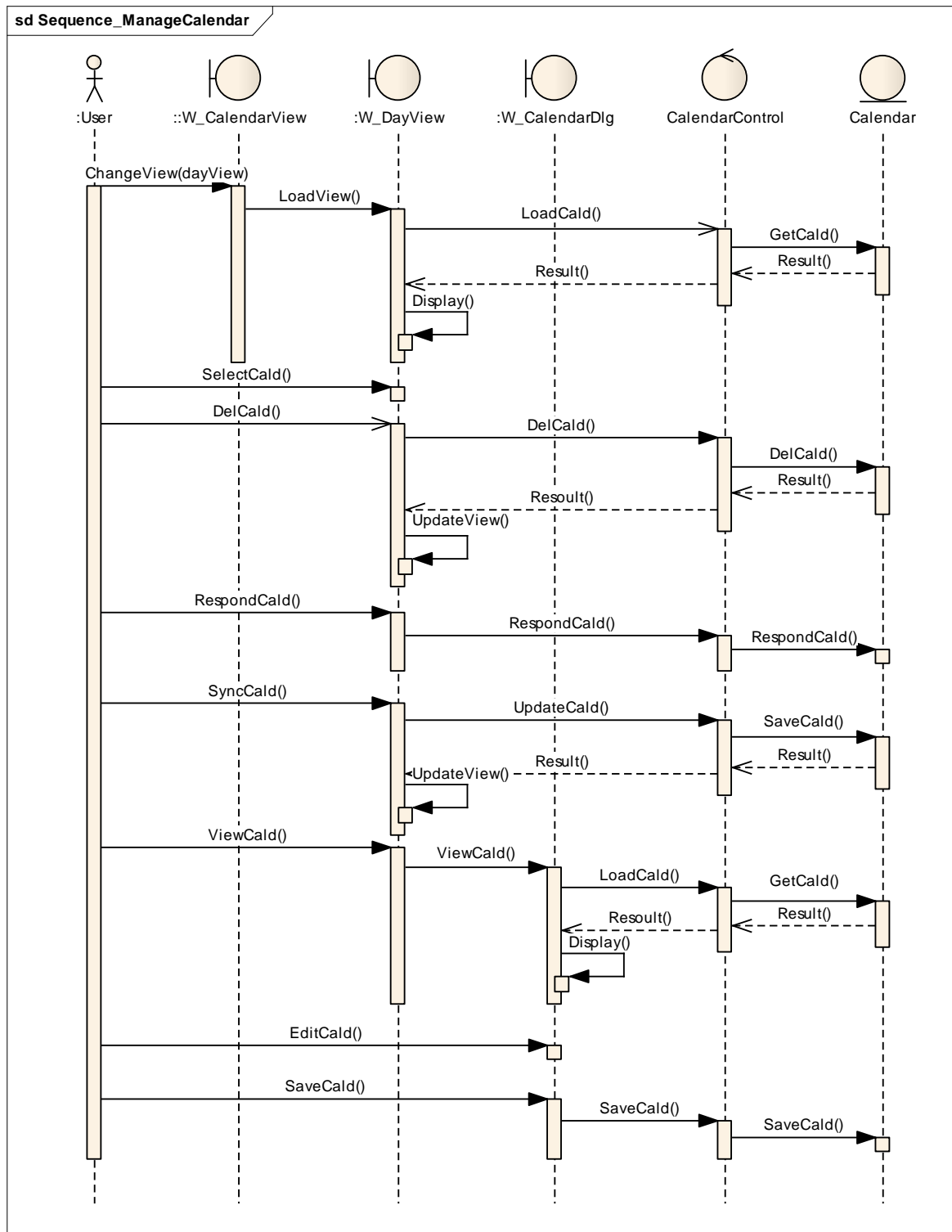
- Hành vi thể hiện trong sự tương tác giữa các đối tượng, tạo nên những kịch bản.
- Hành vi biểu lộ các ứng xử của đối tượng, trước những sự kiện xảy ra đến với nó.
- Hành vi bộc lộ ở công việc và các luồng công việc.

Ở đây ta trình bày hai cách tiếp cận: mô hình hóa tương tác và mô hình hóa ứng xử.

3.4.1 Mô hình hóa sự tương tác

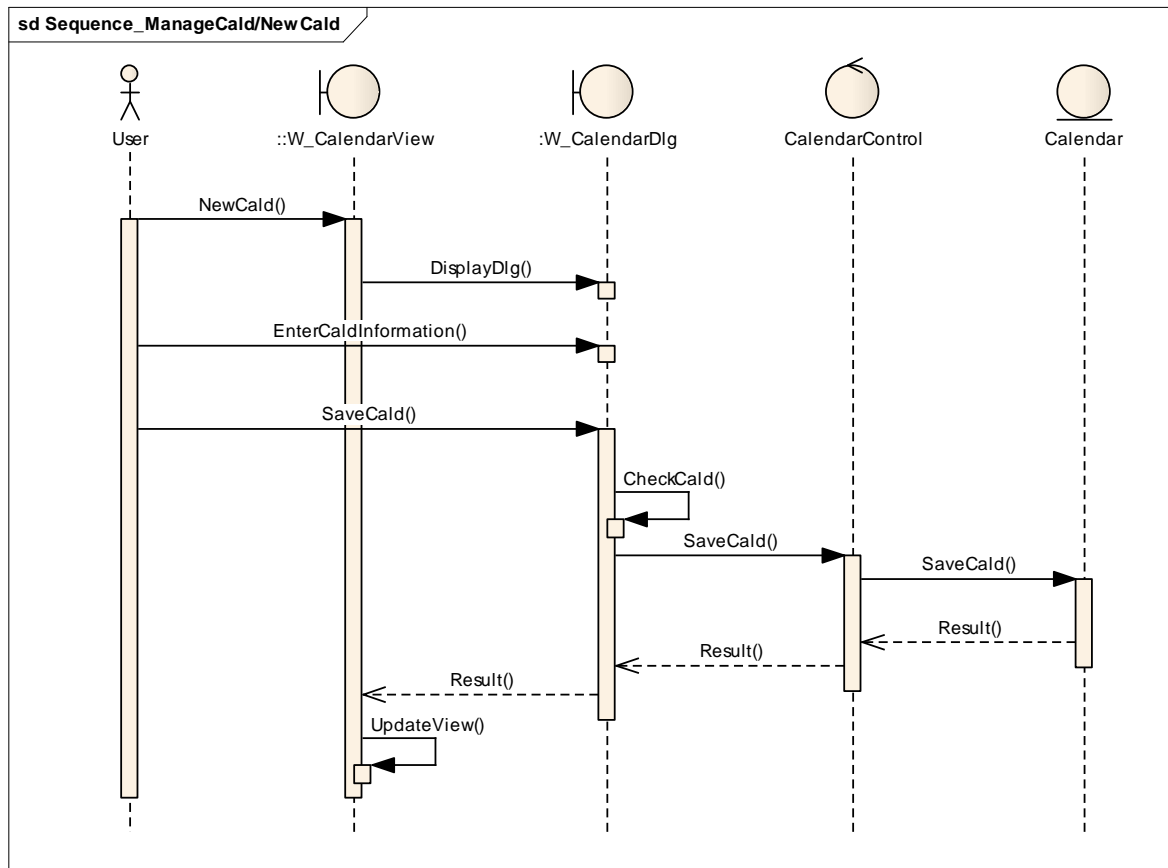
Ở phần trên ta đã trình bày về cấu trúc tĩnh của ca sử dụng bằng các biểu đồ ca sử dụng và chỉ ra các lớp tham gia vào ca sử dụng. Ở phần này ta dùng các biểu đồ tương tác để diễn tả sự tương tác giữa các đối tượng nhằm tạo ra kịch bản của mỗi ca sử dụng của hệ thống, diễn tả khía cạnh động của ca sử dụng.

a) Biểu đồ trình tự cho "Quản lý lịch/Sửa, Xóa lịch"



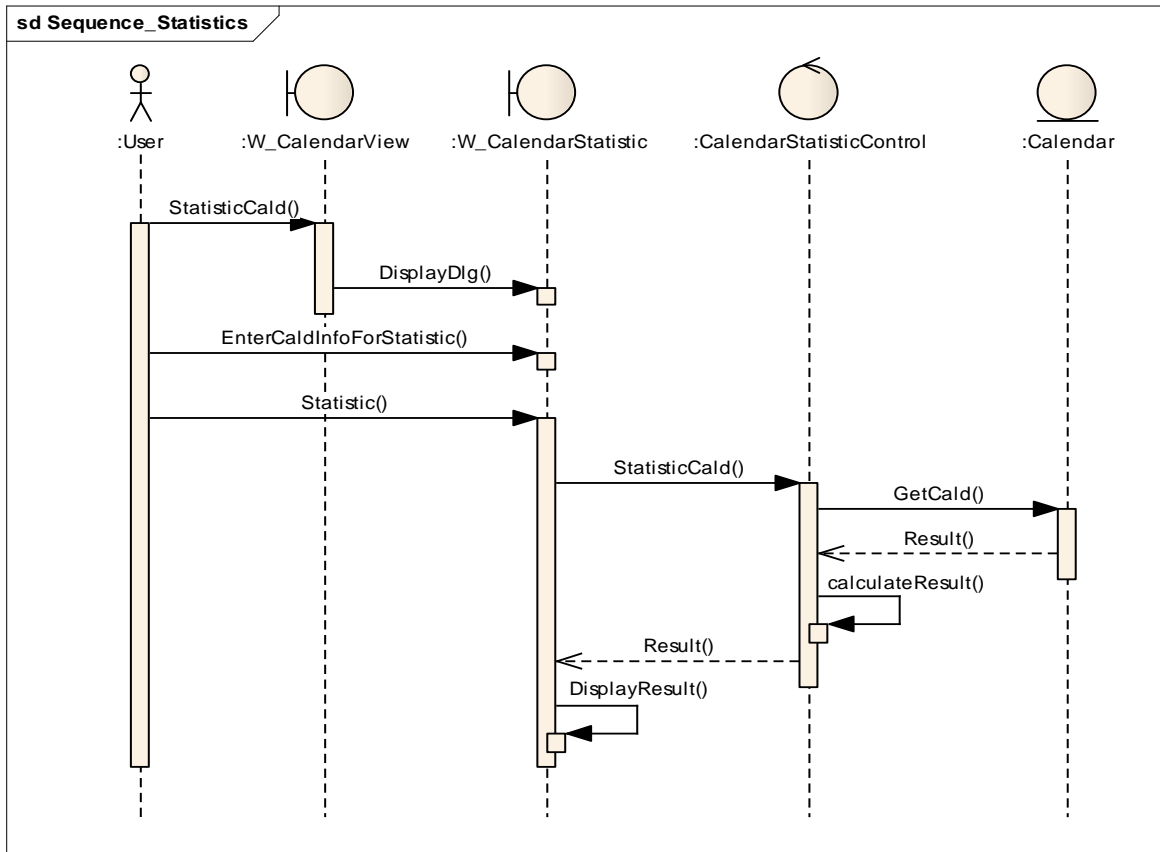
Hình 19. Biểu đồ trình tự cho "Quản lý lịch/Sửa, Xóa lịch"

b) Biểu đồ trình tự cho "Quản lý lịch/Tạo lịch mới"



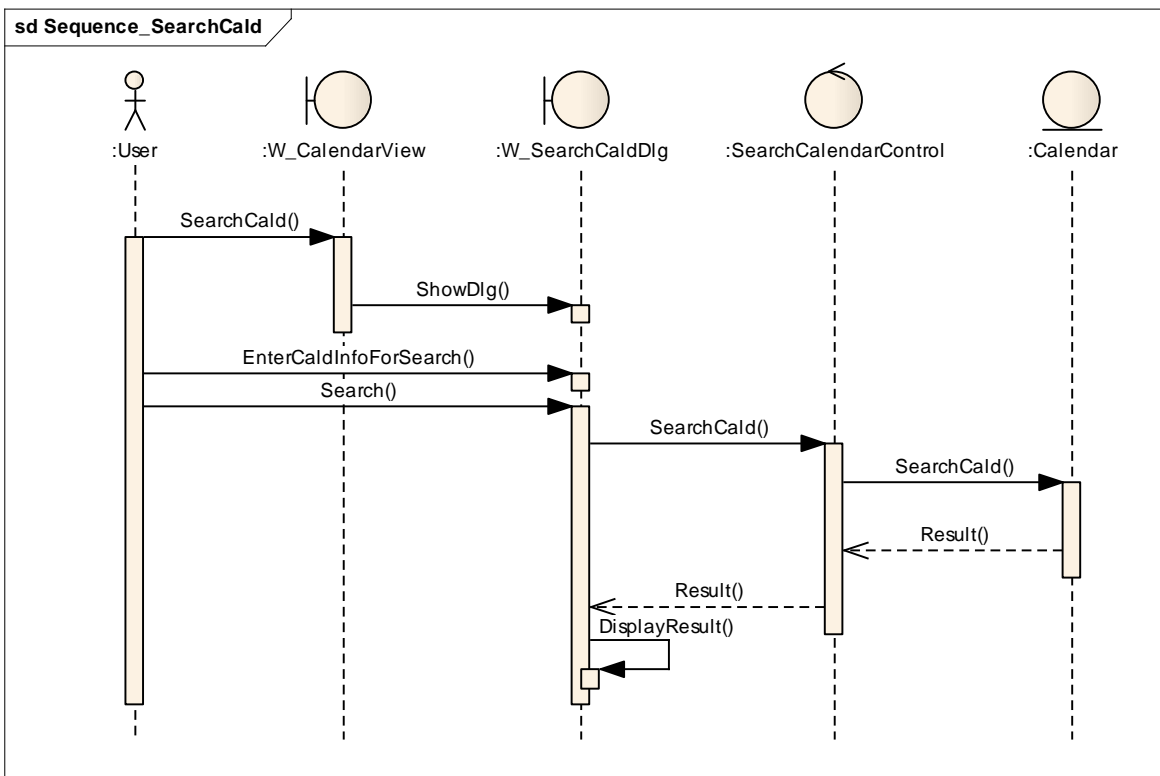
Hình 20. Biểu đồ trình tự cho "Quản lý lịch/Tạo lịch mới"

c) Biểu đồ trình tự cho "Thống kê lịch"



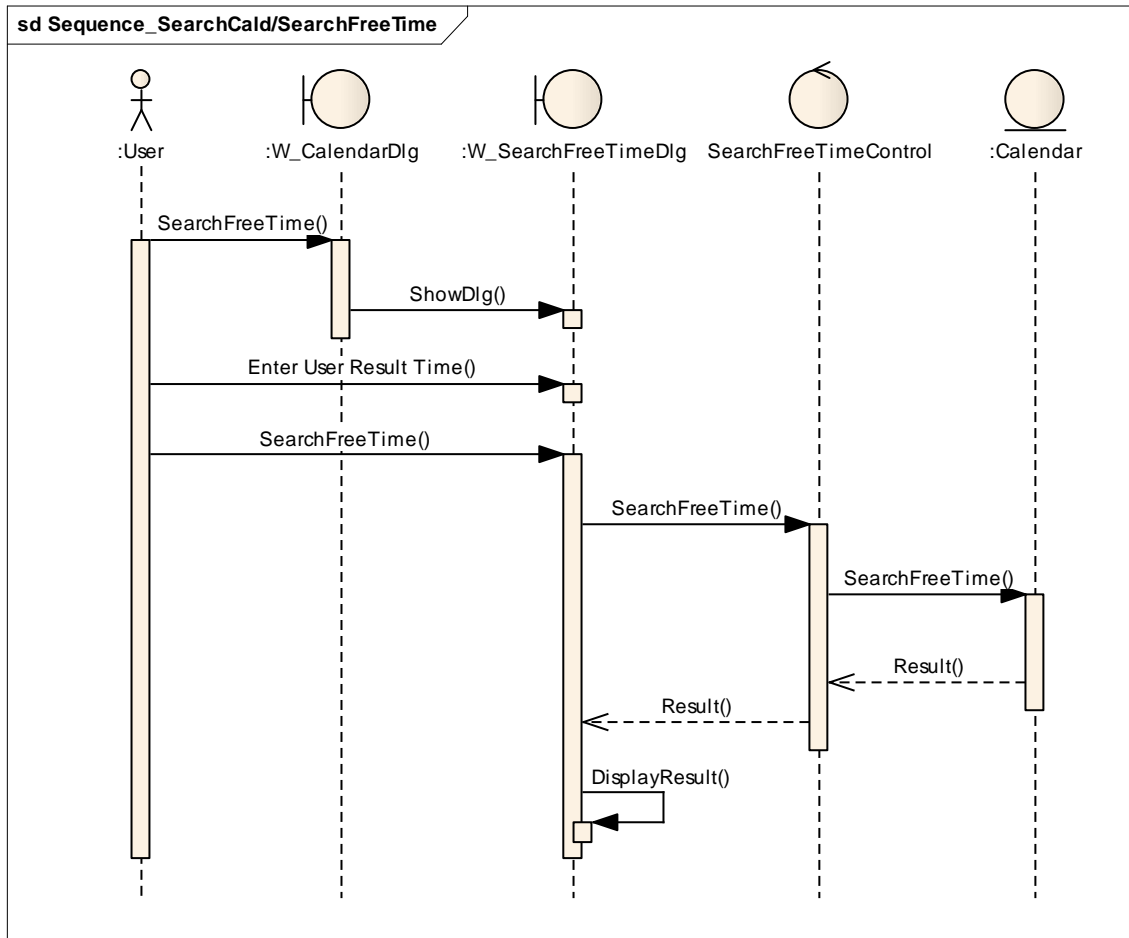
Hình 21. Biểu đồ trình tự cho "Thống kê lịch"

d) Biểu đồ trình tự cho "Tìm kiếm lịch/tìm lịch theo thông tin lịch"



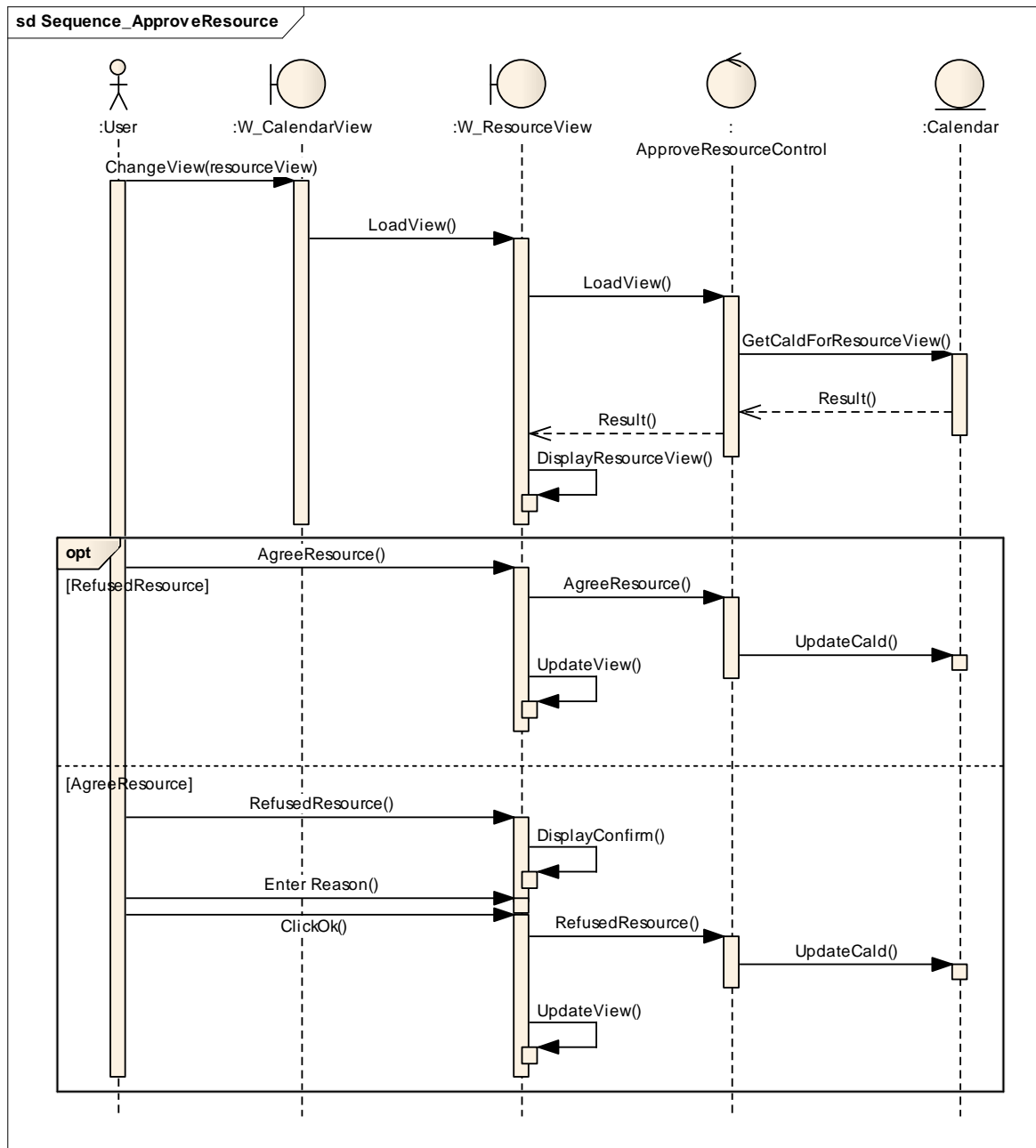
Hình 22. Biểu đồ trình tự cho "Tìm kiếm lịch/tìm lịch theo thông tin lịch"

e) Biểu đồ trình tự cho "Tìm kiếm lịch/Tìm thời gian trống"



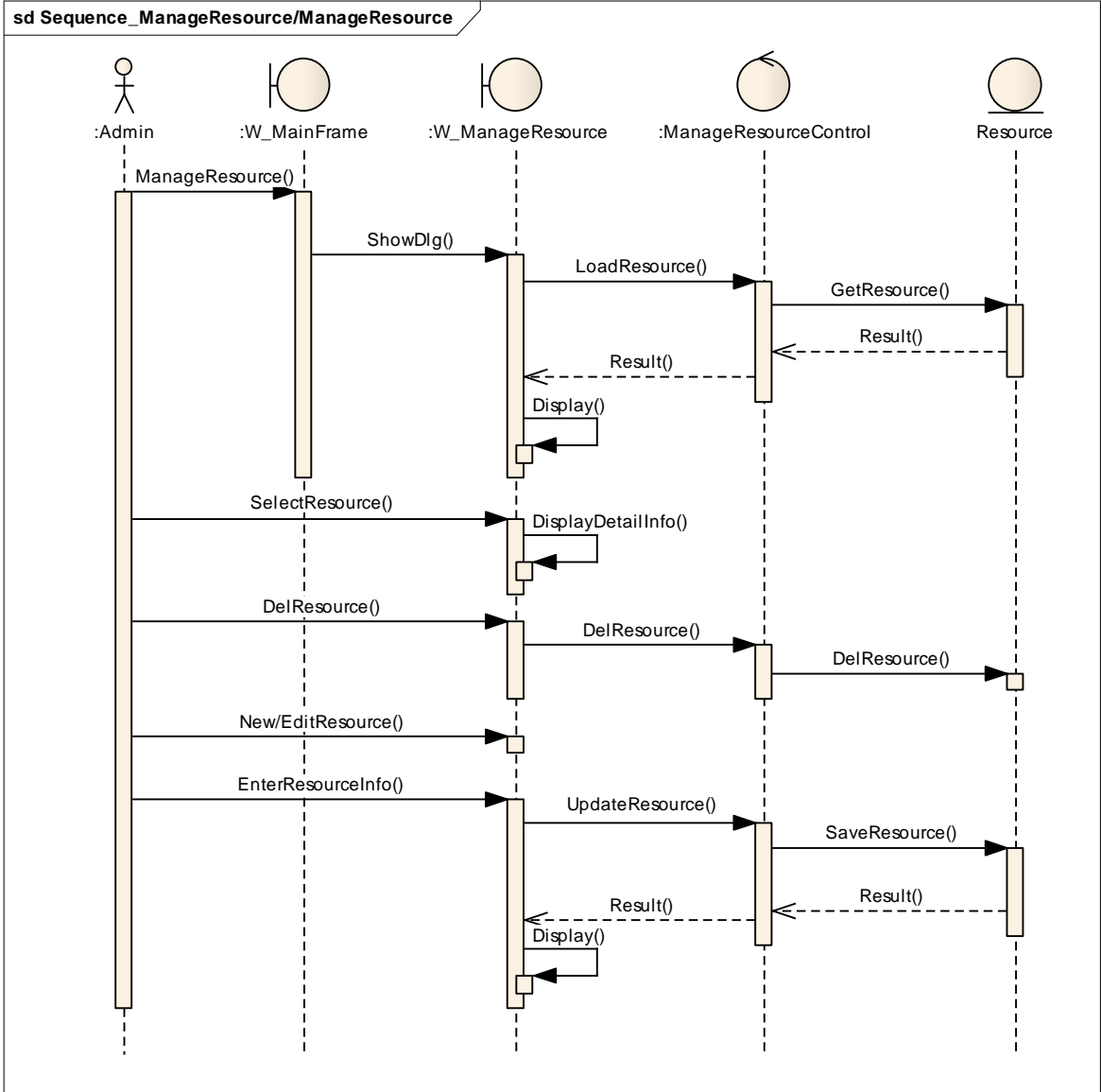
Hình 23. Biểu đồ trình tự cho "Tìm kiếm lịch/Tìm thời gian trống"

f) Biểu đồ trình tự cho "Duyệt tài nguyên"



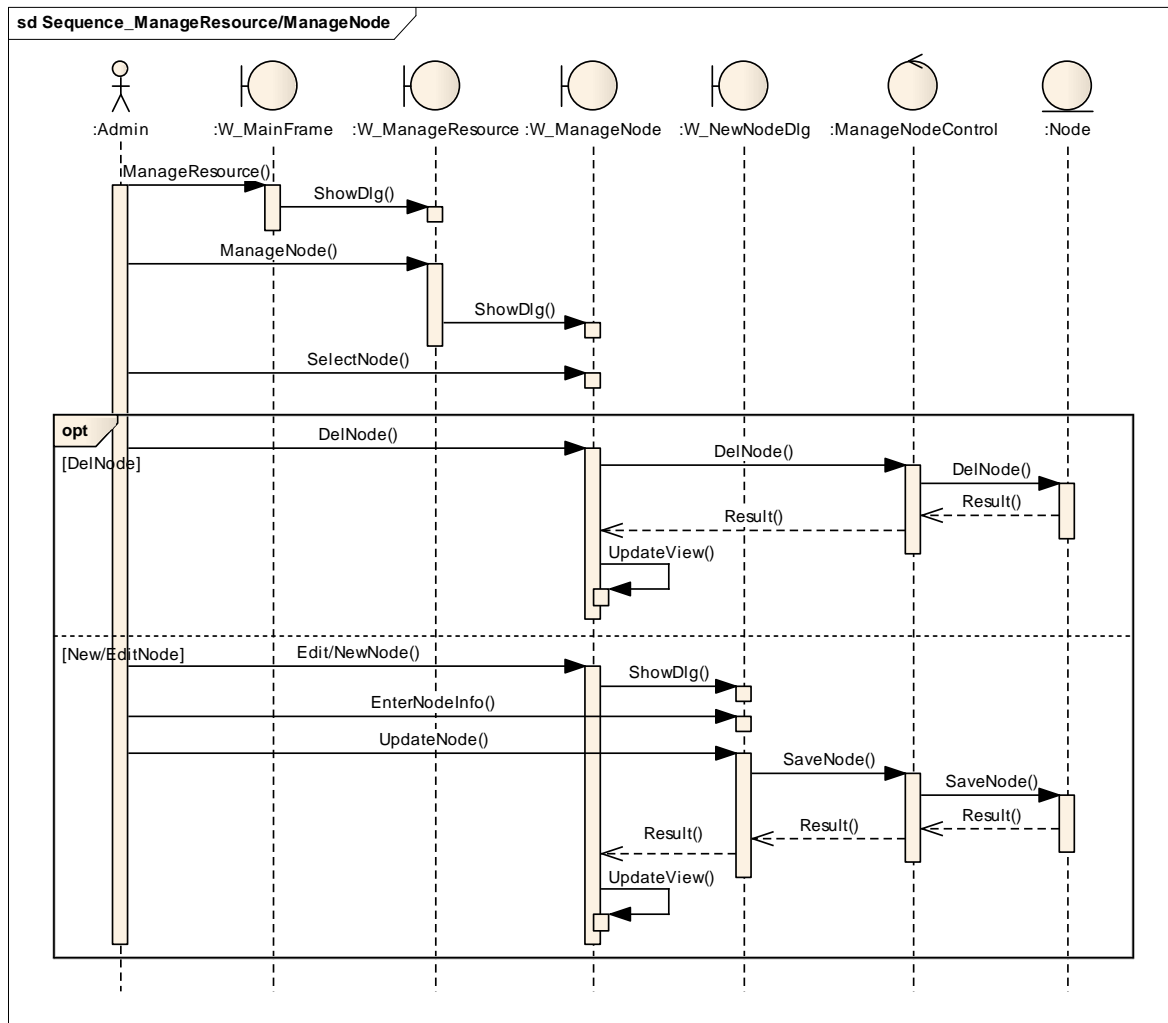
Hình 24. Biểu đồ trình tự cho "Duyệt tài nguyên"

g) *Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý tài nguyên"*



Hình 25. Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý tài nguyên"

h) Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý node"



Hình 26. Biểu đồ trình tự cho "Quản lý tài nguyên/Quản lý node"

CHƯƠNG 4. THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

4.1 Làm nguyên mẫu giao diện người dùng

Trong phần này ta sẽ mô tả các giao diện cần thiết của hệ thống. Muốn làm việc này ta theo dõi từng ca bước trong kịch bản của ca sử dụng, xét nội dung của tương tác giữa đối tác và hệ thống, các thông tin vào và ra, các hành động được yêu cầu để tổ hợp vào một phần tử giao diện. Tiếp theo ta mô tả các phần tử giao diện đã phát hiện và cuối cùng là làm nguyên mẫu của nó bằng các bộ tạo lập giao diện người dùng (GUI builder).

Dưới đây là mô tả các giao diện cần thiết trong các ca sử dụng chính của hệ thống, phần nguyên mẫu của giao diện xem trong phần phụ lục.

4.1.1 Giao diện chính của chương trình

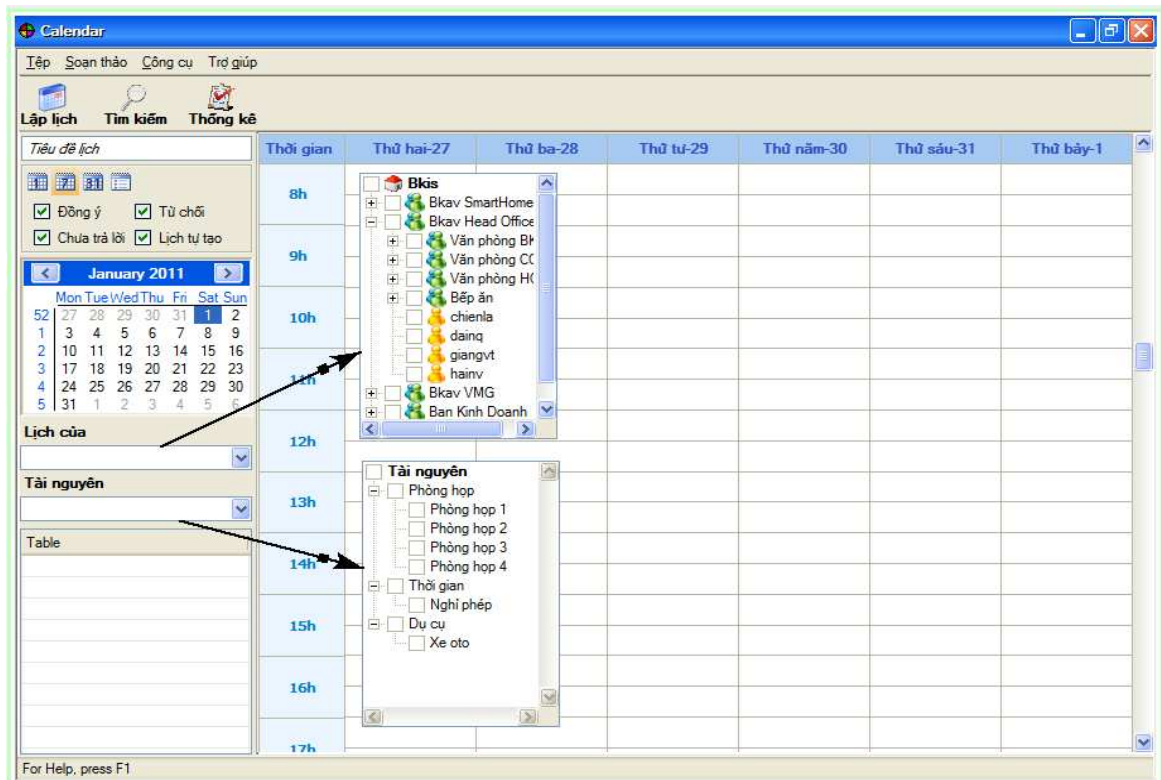
a) Mô tả giao diện

Tên: Giao diện chính
<p><i>Miêu tả ngắn:</i></p> <p>Là giao diện chính của chương trình, trên đây trình bày lịch làm việc theo các kiểu gom nhóm khác nhau. Cho phép người dùng xem và thực hiện được các thao tác với lịch.</p> <p><i>Sử dụng:</i></p> <p>Mỗi khi người dùng đăng nhập vào phần mềm thì hiển thị giao diện này đầu tiên. Đây là giao diện mà người dùng thường xuyên tiếp xúc với.</p> <p><i>Mức độ phức tạp:</i></p> <p>Phức tạp.</p> <p><i>Các trường hiển thị:</i></p> <p>Danh sách các lịch, các lựa chọn để lọc lịch (thành phần tham gia lịch, tài nguyên, kiểu lịch).</p> <p><i>Các khả năng rẽ ngang:</i></p> <p>Thoát chương trình, Chuyển sang các thao tác với lịch, Chuyển sang thao tác tìm kiếm, Chuyển sang thao tác tùy chọn...</p>

Bảng 3. Mô tả giao diện: Giao diện chính

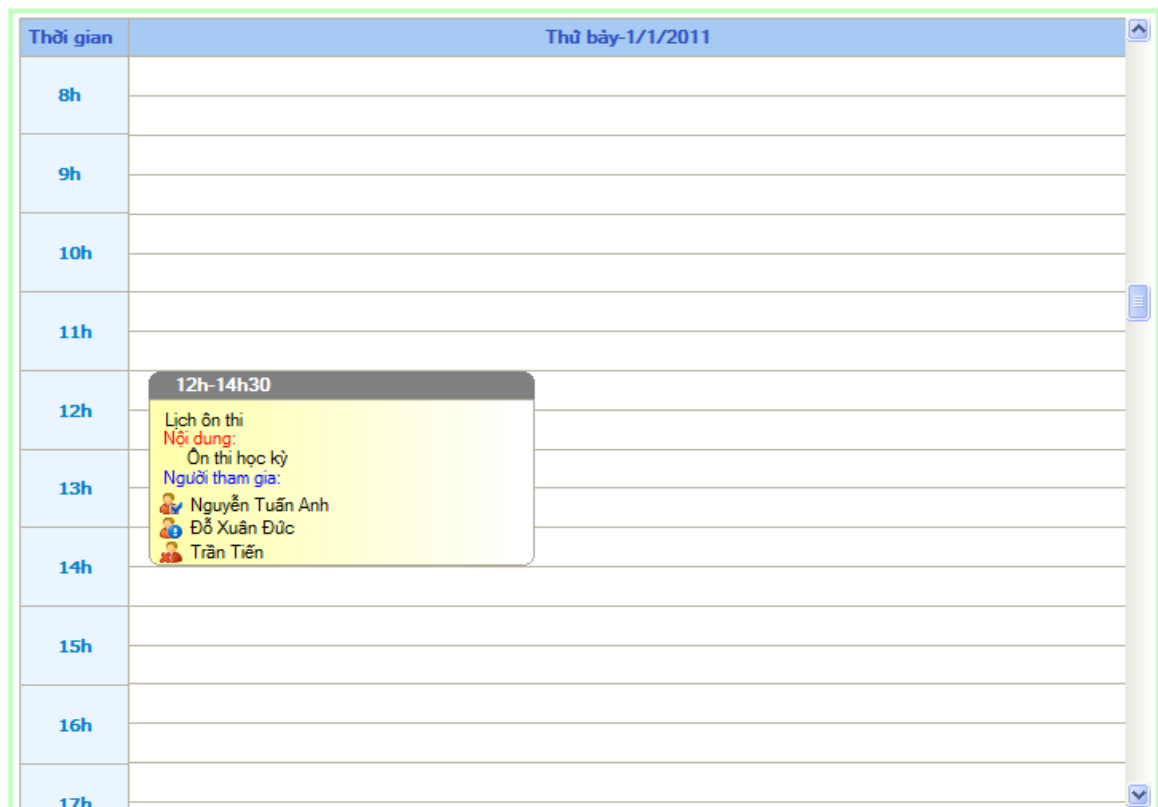
b) Nguyên mẫu giao diện

- Giao diện chính (Xem theo tuần)



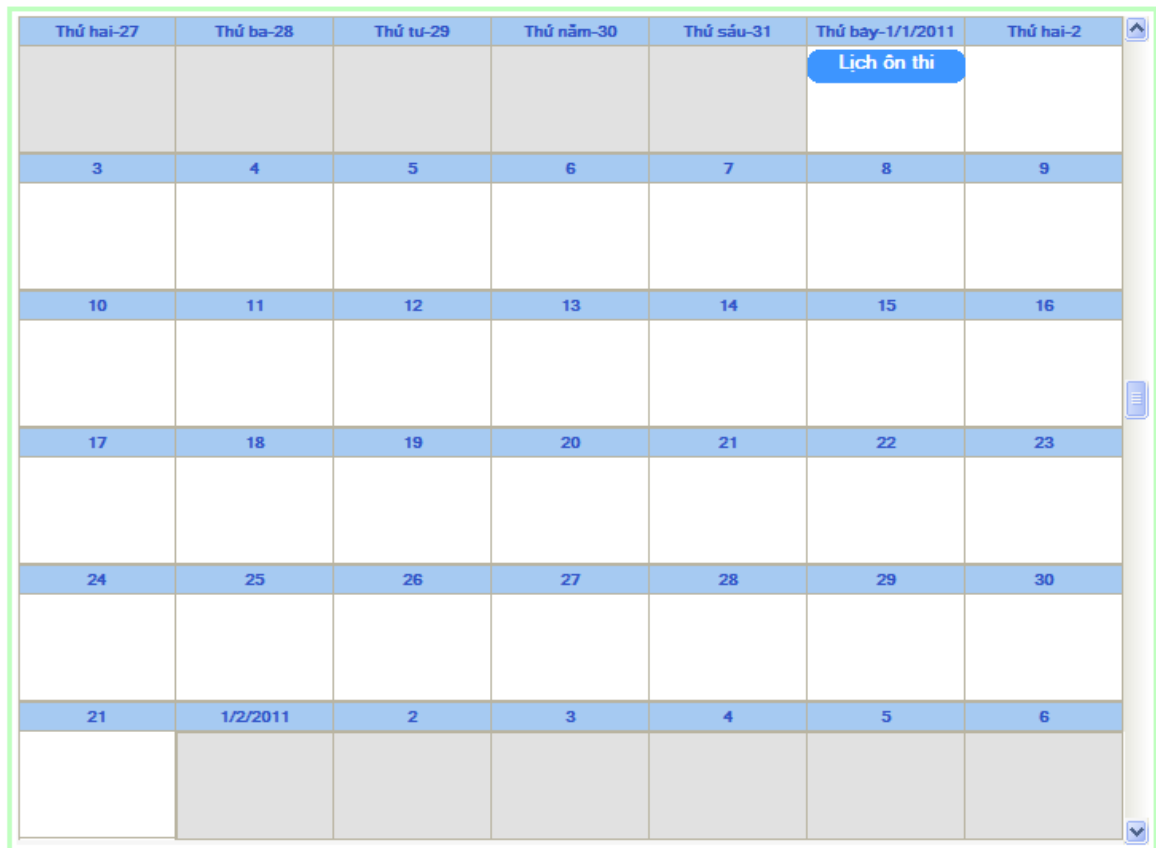
Hình 27. Nguyên mẫu giao diện chính (xem theo tuần)

- Giao diện xem lịch theo ngày



Hình 28. Nguyên mẫu giao diện xem lịch theo ngày

- Giao diện xem lịch theo tháng



Hình 29. Nguyên mẫu giao diện xem lịch theo tháng

4.1.2 Giao diện ca sử dụng ‘Quản lý lịch làm việc’

a) Phân bổ các giao diện cho từng bước trong ca sử dụng

Các bước/ hoạt động trong ca sử dụng	Phần tử giao diện cần có
Tạo một lịch mới Nhập thông tin lịch Thông báo khi lịch bị xung đột	Cửa sổ tạo lịch mới Cửa sổ xung đột lịch
Sửa một lịch	Cửa sổ tạo lịch mới
Xem một lịch	- Cửa sổ tạo lịch mới (đối với trường có quyền chỉnh sửa lịch). - Cửa sổ xem lịch (đối với trường hợp có không có quyền chỉnh sửa lịch).

Bảng 4. Phân bổ các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’

b) *Mô tả giao diện*

- Cửa sổ tạo lịch mới

Tên: Cửa sổ tạo lịch mới
<p><i>Miêu tả ngắn:</i></p> <p>Cho phép người dùng nhập thông tin của một lịch làm việc để tạo hoặc sửa lịch.</p> <p><i>Sử dụng:</i></p> <p>Mỗi khi người dùng tạo lịch mới hoặc sửa lịch thì hệ thống hiển thị cửa sổ này. Đây là giao diện mà người dùng thường xuyên tiếp xúc với.</p> <p><i>Mức độ phức tạp:</i></p> <p>Phức tạp.</p> <p><i>Các trường nhập dữ liệu:</i></p> <p>Các thông tin về lịch như: tên lịch, thời gian lịch diễn ra lịch, địa điểm diễn ra lịch, tài nguyên cần đăng ký, các tùy chọn của lịch...</p> <p><i>Các trường hiển thị:</i></p> <p>Các thông tin liên quan tới lịch.</p> <p><i>Các khả năng rẽ ngang:</i></p> <p>Thoát, Lưu lịch, bỏ qua.</p>

Bảng 5. Mô tả giao diện: Cửa sổ tạo lịch mới

- Cửa sổ xung đột lịch

Tên: Cửa sổ xung đột lịch
<p><i>Miêu tả ngắn:</i></p> <p>Cửa sổ này cho phép người dùng biết các thông tin liên quan tới lịch xung đột khi tạo, sửa một lịch làm việc.</p> <p><i>Sử dụng:</i></p> <p>Khi một lịch bị xung đột về tài nguyên, người tham gia thì cửa sổ này xuất hiện, giúp cho người lập lịch biết được các thông tin xung đột.</p> <p><i>Mức độ phức tạp:</i></p> <p>Chuẩn.</p> <p><i>Các trường hiển thị:</i></p> <p>Danh sách lịch bị xung đột với lịch mới, chỉ rõ được thành phần xung đột trong lịch.</p> <p><i>Các khả năng rẽ ngang:</i></p> <p>Bỏ qua, Sửa lịch, Tạo lịch.</p>

Bảng 6. Mô tả giao diện: Cửa sổ xung đột lịch

- Cửa sổ xem lịch

Tên: Cửa sổ xem lịch
<p>Miêu tả ngắn:</p> <p>Cửa sổ này cho phép người tham gia lịch, người dùng trong hệ thống xem chi tiết một lịch public.</p> <p>Sử dụng:</p> <p>Khi người dùng trong hệ thống muốn xem một lịch public không phải do mình tạo thì cửa sổ này xuất hiện.</p> <p>Mức độ phức tạp:</p> <p>Chuẩn.</p> <p>Các trường hiển thị:</p> <p>Tất cả các thông tin liên quan tới lịch: tên lịch, thời gian lịch diễn ra lịch, địa điểm diễn ra lịch, tài nguyên cần đăng ký, các tùy chọn của lịch.</p> <p>Các khả năng rẽ ngang:</p> <p>Thoát.</p>

Bảng 7. Mô tả giao diện: Cửa sổ xem lịch

4.1.3 Giao diện ca sử dụng ‘Thống kê lịch’

Ca sử dụng này chỉ có một thành phần giao diện là cửa sổ thống kê lịch.

Tên: Cửa sổ thống kê
<p>Miêu tả ngắn:</p> <p>Giao diện này cho phép người dùng thống kê các lịch theo một tiêu chí nào đó như: lịch làm việc trong ngày, tuần, tháng, lịch mình đã tạo...</p> <p>Sử dụng:</p> <p>Người dùng có thể xem danh sách các lịch làm việc trong ngày, tuần...</p> <p>Mức độ phức tạp:</p> <p>Đơn giản.</p> <p>Các trường hiển thị:</p> <p>Danh sách các lịch theo tiêu chí người dùng chọn.</p> <p>Các khả năng rẽ ngang:</p> <p>Kết thúc.</p>

Bảng 8. Mô tả giao diện: Cửa sổ thống kê lịch

4.1.4 Giao diện ca sử dụng ‘Tìm kiếm lịch’

a) Phân bổ các giao diện cho từng bước trong ca sử dụng

Các bước/ hoạt động trong ca sử dụng	Phần tử giao diện cần có
Tìm kiếm theo thông tin lịch	Cửa sổ tìm kiếm lịch
Tìm kiếm thời gian trống	Cửa sổ tìm thời gian trống

Bảng 9. Phân bổ các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’

b) Mô tả giao diện

- Cửa sổ tìm kiếm lịch

Tên: Cửa sổ tìm kiếm lịch
<p><i>Miêu tả ngắn:</i></p> <p>Cho phép người dùng tìm kiếm lịch theo các tiêu chí: tên lịch, thành phần tham gia, khoảng thời gian diễn ra lịch.</p> <p><i>Sử dụng:</i></p> <p>Người dùng có thể tìm lịch làm việc theo tiêu chí nào đó.</p> <p><i>Mức độ phức tạp:</i></p> <p>Chuẩn.</p> <p><i>Các trường nhập dữ liệu:</i></p> <p>Các tiêu chí để tìm kiếm lịch như: tên lịch, thành phần tham gia, khoảng thời gian...</p> <p><i>Các trường hiển thị:</i></p> <p>Danh sách các lịch thỏa mãn điều kiện tìm kiếm, mỗi lịch có thông tin chi tiết.</p> <p><i>Các khả năng rẽ ngang:</i></p> <p>Kết thúc.</p>

Bảng 10. Mô tả giao diện: Cửa sổ tìm kiếm lịch

- Cửa sổ tìm thời gian trống

Tên: Cửa sổ tìm thời gian trống
<p><i>Miêu tả ngắn:</i></p> <p>Cho phép người tạo lịch tìm một khoảng thời gian mà khi đó người hay tài nguyên có thể tham gia lịch.</p> <p><i>Sử dụng:</i></p> <p>Trước hoặc trong khi tạo lịch người tạo lịch có thể tìm khoảng thời gian trống của tài nguyên, người dùng khác. Giúp cho việc lập lịch nhanh chóng hơn và không bị xung đột với các lịch đã được tạo trước đó.</p>

<p><i>Mức độ phức tạp:</i></p> <p>Chuẩn.</p> <p><i>Các trường nhập dữ liệu:</i></p> <p>Danh sách người dùng, danh sách tài nguyên, thời gian diễn ra, khoảng thời gian dự kiến của lịch.</p> <p><i>Các trường hiển thị:</i></p> <p>Danh sách các khoảng thời gian mà khi đó người dùng, tài nguyên có thể sử dụng được (chưa tham gia vào một lịch nào đó).</p> <p><i>Các khả năng rẽ ngang:</i></p> <p>Kết thúc, đồng ý.</p>

Bảng 11. Mô tả giao diện: Cửa sổ tìm thời gian trống

4.1.5 Giao diện ca sử dụng ‘Duyệt tài nguyên’

a) Phân bố các giao diện cho từng bước trong ca sử dụng

Các bước/ hoạt động trong ca sử dụng	Phần tử giao diện cần có
Duyệt tài nguyên	Màn hình chính duyệt tài nguyên
Đồng ý tài nguyên	
Từ chối tài nguyên	Cửa sổ từ chối tài nguyên

Bảng 12. Phân bố các giao diện cho các bước trong ca sử dụng ‘Quản lý lịch làm việc’

- *Mô tả giao diện*

Tên: Cửa sổ tìm kiếm lịch
<p><i>Miêu tả ngắn:</i></p> <p>Màn hình duyệt tài nguyên cho phép người duyệt tài nguyên chấp nhận hay từ chối một lịch liên quan quan tới tài nguyên mình có quyền duyệt một cách nhanh chóng.</p> <p><i>Sử dụng:</i></p> <p>Người duyệt tài nguyên có thể xem danh sách các lịch liên quan tới tài nguyên mình có quyền duyệt và duyệt những lịch này.</p> <p><i>Mức độ phức tạp:</i></p> <p>Chuẩn.</p> <p><i>Các trường hiển thị:</i></p> <p>Danh sách các lịch có tài nguyên cần duyệt.</p>

Bảng 13. Mô tả giao diện: Cửa sổ tìm kiếm lịch

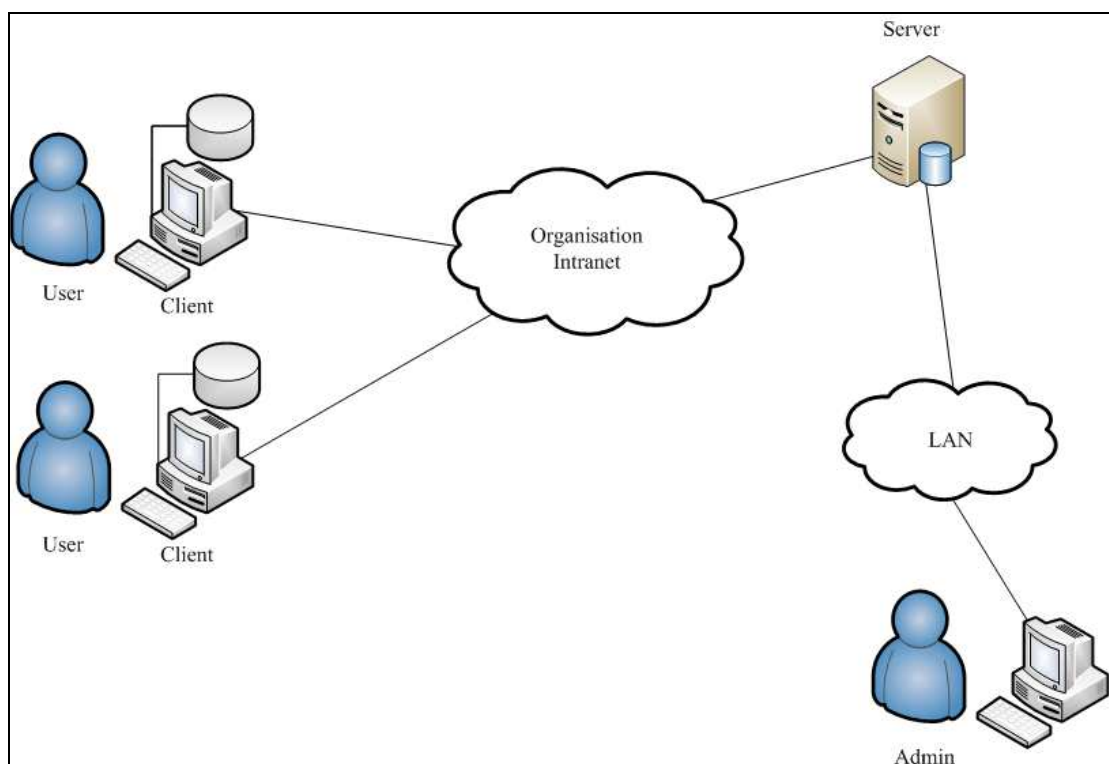
4.2 Thiết kế hệ thống

Dựa trên kiến trúc chương trình và phân tích chức năng ta chia hệ thống thành 2 phần riêng biệt: Client và Server

- Server có nhiệm vụ nhận các yêu cầu từ client, như xử lý đăng nhập, yêu về dữ liệu, đồng bộ... và trả về kết quả xử lý cho client. Ngoài ra, quản trị hệ thống có thể có thao tác với hệ thống như thêm người dùng, thêm tài nguyên...
- Client gồm các module tương tác trực tiếp với người dùng. Nó bảo đảm đầy đủ các tính năng như đã phân tích.

Hệ thống được triển khai trên nền tảng mạng LAN. Server sẽ chịu trách nhiệm đáp ứng nhu cầu của người dùng. Mỗi người dùng ở các máy client có một tài khoản để đăng nhập vào hệ thống. Dữ liệu, thao tác của người dùng được đồng bộ với server. Người quản trị quản trị hệ thống bằng các remote vào máy tính cài server để cấu hình, quản trị tài nguyên, người dùng.

Dưới đây là mô hình bố trí vật lý của hệ thống.



Hình 30. Mô hình bố trí vật lý của hệ thống

4.3 Thiết kế chi tiết

Trong các phần trên, ta tập trung vào nghiên cứu cấu trúc logic của các thông tin, cần thiết cho việc xây dựng một giải pháp nghiệp vụ. Trong phần thiết kế chi tiết này ta nghiên cứu cách tốt nhất để cài đặt cấu trúc logic nói trên, nhằm tối ưu hóa hiệu năng của ứng dụng.

Phần mềm cần phải đáp ứng các nhu cầu về tốc độ và chiếm ít tài nguyên của hệ thống. Ngoài ra phần mềm có giao diện đẹp khả năng mở rộng tốt. Ngôn ngữ VC++ đáp ứng đầy đủ tiêu chí trên, hơn nữa VC++ hỗ trợ lập trình mạng thích hợp với chương trình hoạt động theo mô hình client-server nên VC++ đã được chọn để phát triển hệ thống.

Môi trường phát triển phần mềm là bộ công cụ lập trình Visual Studio 2010 của Microsoft, đây là IDE mạnh hỗ trợ toàn diện cho lập trình với ngôn ngữ VC++.

Vấn đề cơ sở dữ liệu sẽ thực hiện như sau: trên server, sử dụng hệ quản trị cơ sở dữ liệu bằng SQL server 2005, nhằm làm tăng khả năng đáp ứng cho nhiều yêu cầu từ nhiều người dùng cùng kết nối tới server. Ở phía client sẽ dùng hệ quản trị cơ sở dữ liệu SQLite. Lý do chọn SQLite vì SQLite là một hệ quản trị cơ sở dữ liệu nhỏ gọn, không cần cài đặt, nó còn hỗ trợ SQL92 hơn thế nữa SQLite còn hoàn toàn miễn phí.

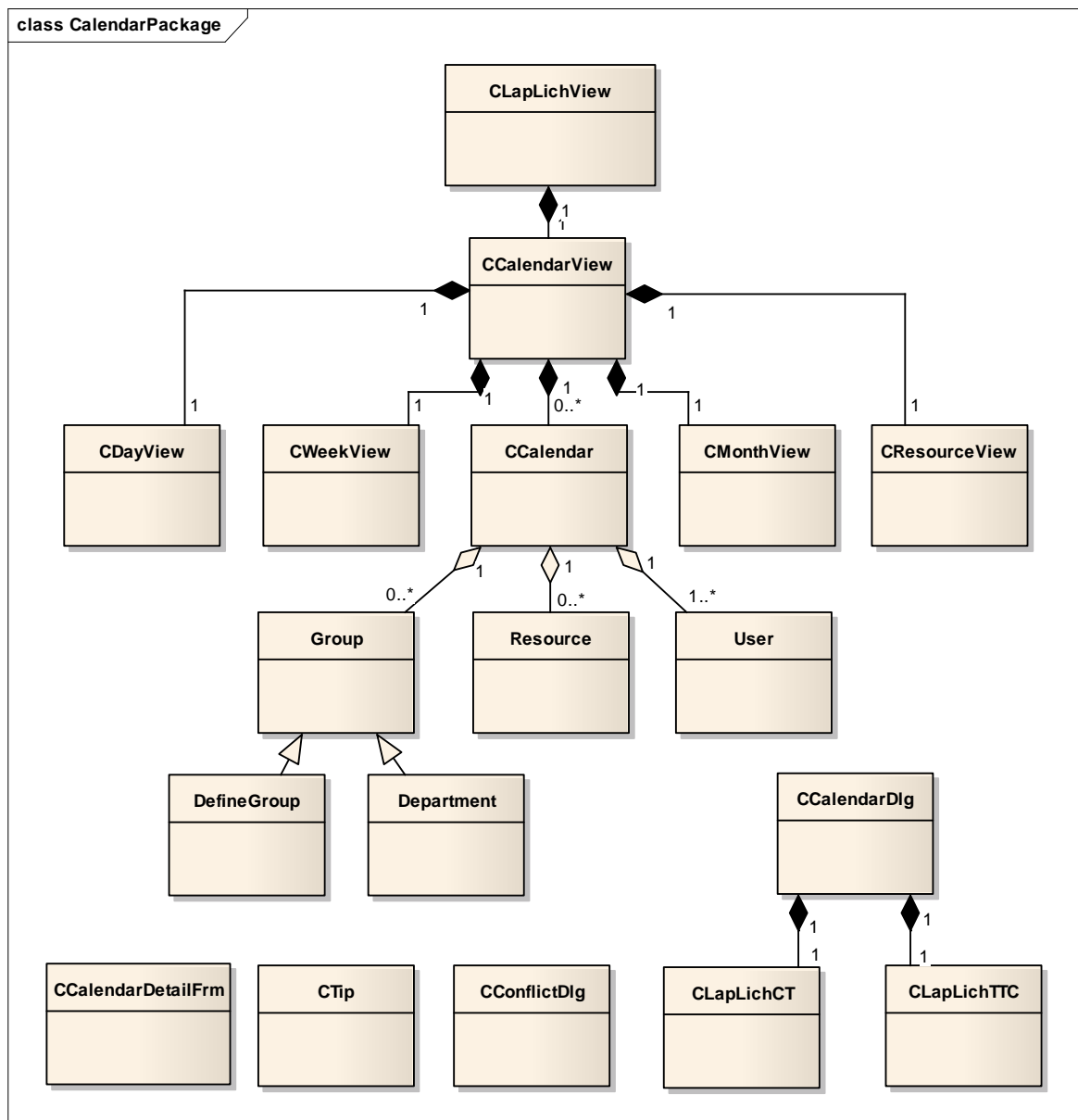
4.3.1 Biểu đồ lớp thiết kế

Lớp thiết kế là những lớp được tạo là dựa trên lớp phân tích ở trên khi đã thêm vào các yếu tố liên quan đến ngôn ngữ lập trình, môi trường phát triển hệ thống; chính nó sẽ được chuyển đổi một cách dễ dàng thành chương trình.

Chương trình ở client được chia thành ba phần chính: Phần lập lịch với các lớp thao tác, quản lý lịch làm việc, phần socket gồm các lớp truyền nhận dữ liệu với server qua mạng, phần thao tác cơ sở dữ liệu gồm các lớp thao tác với cơ sở dữ liệu SQLite.

a) Một số lớp thuộc gói lập lịch

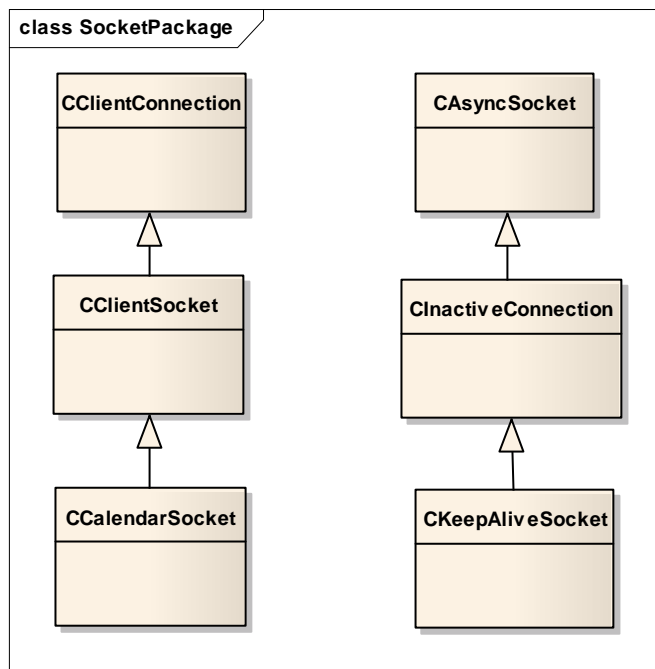
- Lớp CLapLichView: Là lớp giao diện chính nó tích hợp các thành phần giao diện khác. Xử lý các sự kiện như chọn người tham gia, tài nguyên...
- Lớp CcalendarView: Là lớp thể hiện các góc nhìn khác nhau của lịch làm việc, nó bao gồm các kiểu nhìn như: xem theo ngày, xem theo tuần, tháng...
- Các lớp User, Resource, Group, Department, DefineGroup: là những lớp lưu trữ thông tin về người dùng, nhóm người dùng, phòng ban.
- Lớp CCalendarDlg: Lớp hiển thị chi tiết nội dung lịch, nó cho phép sửa đổi thông tin của lịch.
- Lớp CCalendarDetailFrm: Là lớp hiển thị chi tiết thông tin khi xem lịch, nó không cho phép sửa đổi thông tin lịch.
- Lớp CTip: là lớp thể hiện thông tin mô tả của lịch khi người dùng di chuột qua.
- Lớp CConflictDlg: là lớp giao diện thông báo xung đột lịch.



Hình 31. Biểu đồ lớp thiết kế thuộc gói lập lịch

b) Các lớp thuộc gói socket

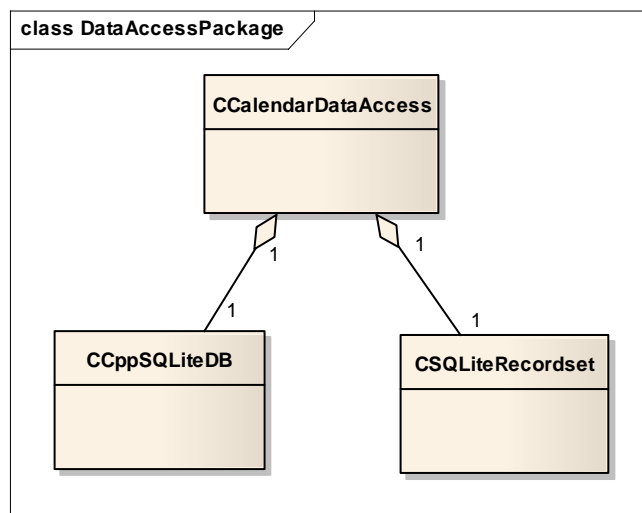
- Lớp **CInactiveConnection**: Kế thừa từ lớp MFC **CAsyncSocket**, dùng để xử lý gửi, nhận và đóng socket. Lớp này thực hiện trả về các Tag, các session.
- Lớp **CKeepAliveSocket**: Thực hiện giữ kết nối, phân tích cấu trúc câu lệnh, và trả về các lệnh để xử lý.
- Lớp **CClientConnection**: Là lớp giao diện. Thực hiện một số việc gửi nhận dữ liệu, phân tích câu lệnh.
- Lớp **CCalendarSocket**: Thực hiện các lệnh trao đổi dữ liệu giữa client và server.



Hình 32. Biểu đồ lớp thiết kế thuộc gói Socket

c) Các lớp thuộc gói DataAccess

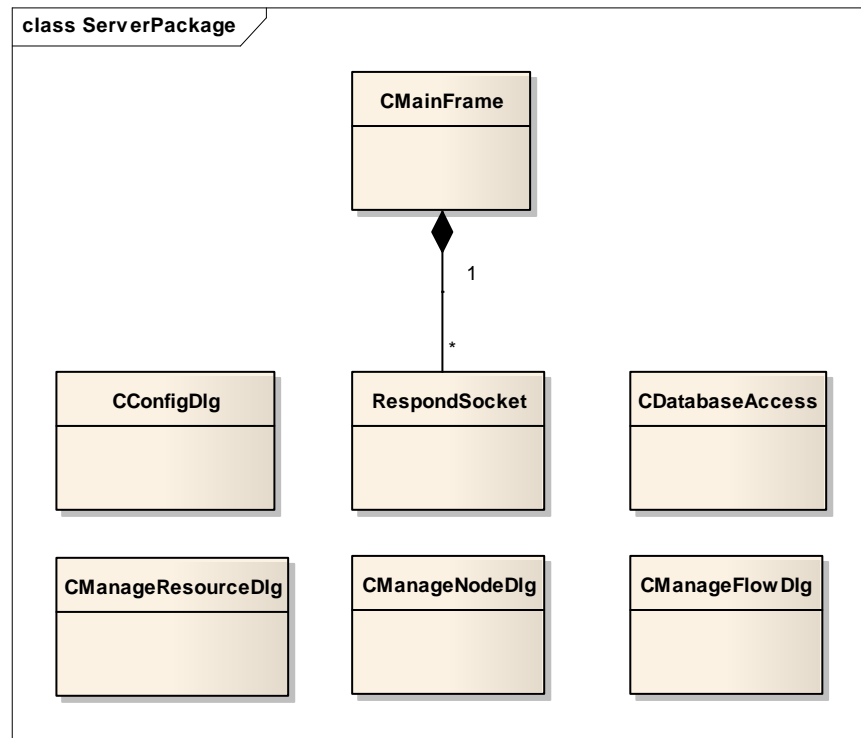
- Lớp CalendarDataAccess: Là lớp chứa các hàm thao tác với cơ sở dữ liệu và trả về kết quả. Trong lớp này sử dụng hai lớp Wrapper của các hàm trong thư viện SQLite để thực thi việc thao tác với cơ sở dữ liệu.
- Lớp CCppSQLiteDB và lớp CSQLiteRecordset: là hai lớp kết nối đến cơ sở dữ liệu SQLite. Cung cấp các hàm xử lý đối với cơ sở dữ liệu.



Hình 33. Biểu đồ lớp thiết kế thuộc gói 'Thao tác dữ liệu'

d) Biểu đồ lớp của server

Gói server bao gồm lớp thể hiện giao diện chính, lớp giao diện hỗ trợ người sử dụng config các thông số như máy chủ database, tài khoản đăng nhập vào database, các giao diện quản trị tài nguyên..., lớp socket xử lý các yêu cầu và phản hồi cho client, và lớp truy cập vào cơ sở dữ liệu.



Hình 34. Biểu đồ lớp thiết kế thuộc gói server

4.3.2 Mô hình trao đổi giữa client-server

Dựa trên quá trình tìm hiểu giải pháp cho mô hình client-server, việc đặt server hoạt động theo cơ chế IOCP là thích hợp nhất do cơ chế IOCP hỗ trợ khả năng đáp ứng được số lượng kết nối lớn và có hiệu năng cao.

Client hoạt động theo chế độ nonblocking, trong quá trình thực hiện để hệ thống được real time thì client phải tạo một socket liên tục nhận dữ liệu từ server gửi về rồi cho vào hàng đợi, thông tin được lấy từ hàng đợi này ra để xử lý, dùng buffer chung.

Giao thức trao đổi của hệ thống

Khi một client muốn tham gia vào hệ thống phải kết nối với server thông một socket (gồm địa chỉ IP và cổng). Mỗi lệnh trao đổi giữa client và server phải tuân theo cấu trúc riêng trong giao thức. Trước mỗi lệnh phải có ID của lệnh đó hay gọi là <Tag> để đảm bảo không bị sai, vì server có thể trả về không theo thứ tự lệnh gửi lên nên trước mỗi kết quả trả về đều có ID của lệnh gửi lên tương ứng.

Server có thể tự động trả kết quả về cho client mà không cần có lệnh yêu cầu của client, khi này <Tag> trả về luôn luôn là 0. Đây là các thông tin mà server chủ động gửi về như admin gửi thông báo cho user, phiên bản mới được update,...

Lệnh trao đổi giữa client-server có dạng:

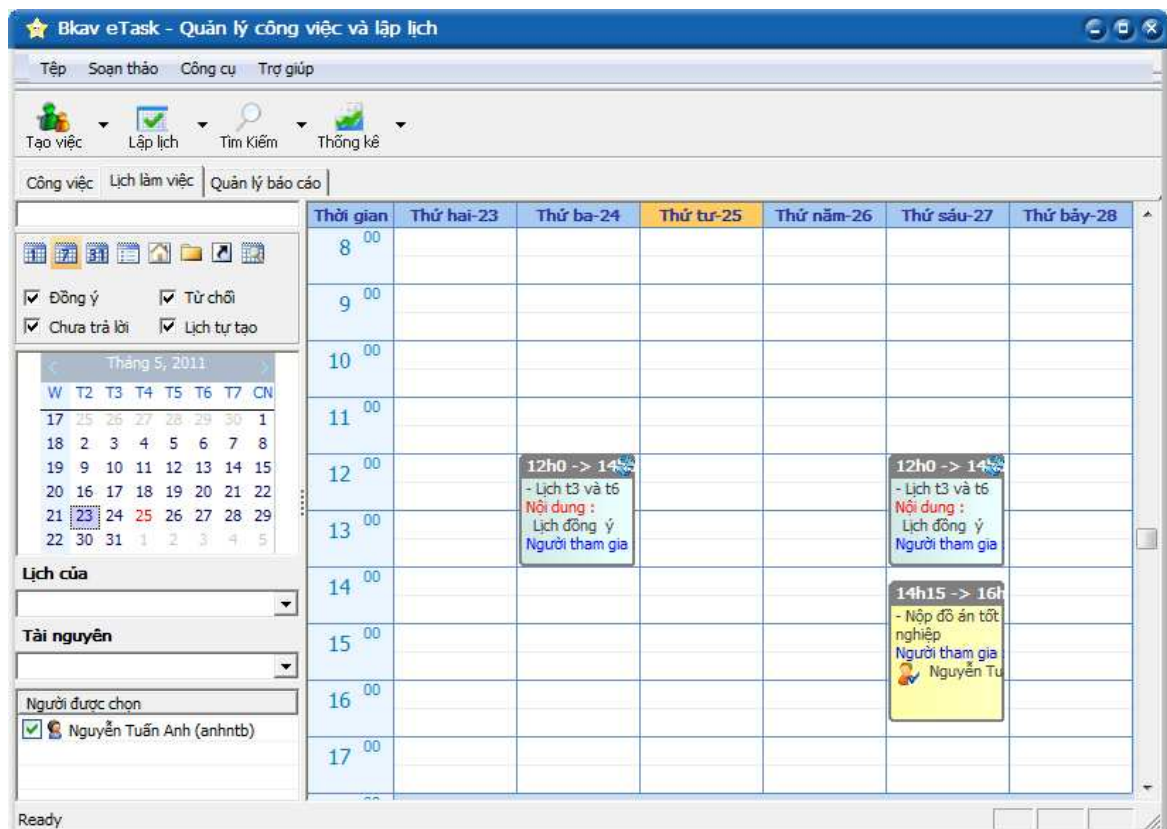
<Tag><SP><Command_code><SP><Header's Length><\r\n>[<Data>]

4.4 Kết quả đạt được

Sau quá trình cài đặt hệ thống đã thực hiện được các chức năng chính như: quản lý lịch làm việc, nhắc lịch, quản lý tài nguyên.

Sau đây là một vài hình ảnh về chương trình.

- Giao diện chính:



Hình 35. Giao diện chính

- Giao diện tạo lịch mới:

Hình 36. Cửa sổ tạo lịch mới

So sánh với các sản phẩm cùng loại, phần mềm có những ưu nhược điểm như sau:

- Phần mềm được xây dựng là một ứng dụng desktop, nên đã khắc phục được các nhược điểm của các sản phẩm cùng loại trên nền web như: tốc độ xử lý nhanh hơn, tương tác với người dùng linh hoạt hơn. Nhưng nó cũng có nhược điểm là máy người dùng phải cài đặt và kết nối vào mạng nội bộ của công ty thì mới sử dụng được.
- So với các sản phẩm cùng loại trên nền desktop, phần mềm đã được bổ sung thêm các chức năng phục vụ cho việc đa người dùng.
- Phần mềm có ưu điểm là ngôn ngữ hoàn toàn tiếng Việt, giao diện đơn giản, dễ sử dụng phù hợp với người dùng Việt.

PHẦN KẾT LUẬN

CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

Qua thời gian nghiên cứu, tìm hiểu và hoàn thành đề tài, bản thân em đã thu được rất nhiều kiến thức bổ ích trong việc xây dựng và phát triển phần mềm, từ khâu khảo sát yêu cầu người dùng cho đến khâu cài đặt và bảo trì. Bên cạnh đó, em cũng đã học hỏi thêm được rất nhiều nội dung bổ ích trong một số lĩnh vực riêng của ngành công nghệ thông tin, cụ thể là:

- Các kiến thức về lập trình mạng và lập trình đa luồng trên Windows.
- Cơ chế làm việc của mô hình client-server.
- Khảo sát, phân tích và thiết kế một hệ thống theo phương pháp hướng đối tượng.

Với lượng kiến thức thu được, em đã có thể tự tin hơn trước những bài toán mới trong tương lai.

Về mặt sản phẩm, hệ thống được cài đặt và bước đầu đưa vào dùng thử và đáp ứng được các nhu cầu cơ bản của người dùng:

- Có các chức năng cơ bản hỗ trợ tạo lịch làm việc một cách nhanh chóng
- Hoạt động trên môi trường mạng nội bộ với nhiều người cùng tham gia.
- Giao diện thân thiện với ngôn ngữ tiếng Việt.

Bên cạnh những kết quả đã đạt được, do kinh nghiệm còn chưa nhiều, nên phần mềm cũng không tránh khỏi những thiếu sót nhất định như giao diện chưa bắt mắt, chưa hỗ trợ đa ngôn ngữ, cơ chế thông báo (khi có sự thay đổi lịch) chưa hợp lý. Các tính năng này sẽ được cải thiện trong thời gian tới.

5.2 Định hướng phát triển

Phần mềm sẽ tiếp tục được phát triển theo hướng hoàn thiện và bổ sung thêm các tính năng cho phù hợp với nhu cầu của người dùng. Trong thời gian tới, phần mềm được định hướng như sau:

- Tiếp tục hoàn thiện và tối ưu các chức năng đã và đang thực hiện
- Đưa ra phiên bản dùng thử. Tích cực nhận các ý kiến đóng góp của người dùng để không ngừng cải tiến sản phẩm
- Tìm hiểu và thêm module mã hóa dữ liệu trong việc lưu trữ và truyền nhận dữ liệu qua mạng, nhằm bảo đảm tính an toàn và bảo mật.
- Phát triển một phiên bản trên nền web, phiên bản này thuận tiện cho những người phải đi công tác, không có điều kiện vào mạng nội bộ của công ty.

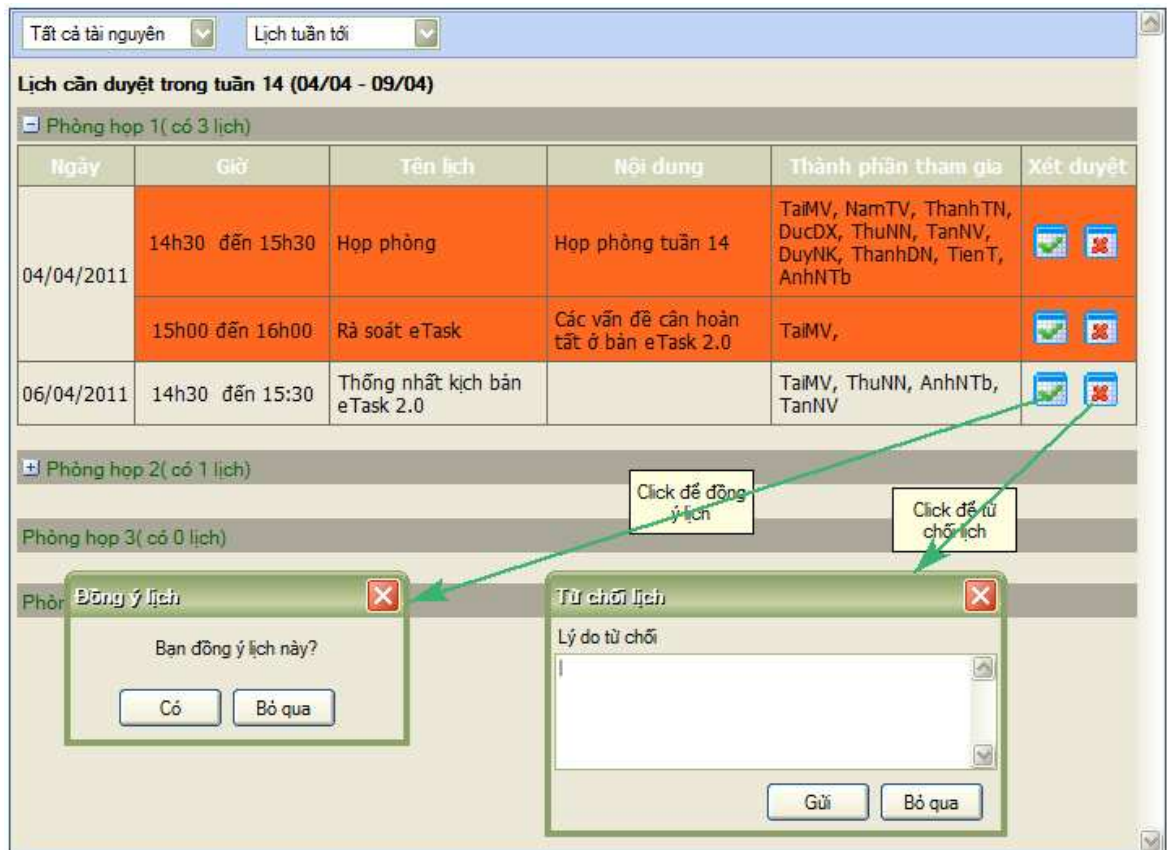
TÀI LIỆU THAM KHẢO

1. Nguyễn Văn Ba, Phát triển hệ thống hướng đối tượng với UML 2.0 và C++, tái bản lần 2, Nhà xuất bản đại học quốc gia Hà Nội, 2008.
2. TS. Dương Kiều Hoa – Tôn Thất Hòa, Phân tích thiết kế hệ thống theo UML, ebook.
3. David Chapman, Teach yourself VC++ 6.0 in 21 days, SAMS publishing, 1998.
4. Anthony Jones and Jim Ohlund, Network Programming for Microsoft Windows, Microsoft Press, 1999.
5. Microsoft Corporation, Desktop Applications with Microsoft Visual C++ 6.0, Microsoft Press, 1999.
6. <http://www.codeguru.com> lần truy cập cuối 5/2011.
7. <http://vi.wikipedia.org/wiki/Client-server> lần truy cập cuối 5/2011.
8. <http://msdn.microsoft.com> lần truy cập cuối 5/2011.
9. <http://www.codeproject.com> lần truy cập cuối 5/2011.

PHỤ LỤC

1 Các nguyên mẫu giao diện của ứng dụng

a) Màn hình duyệt tài nguyên



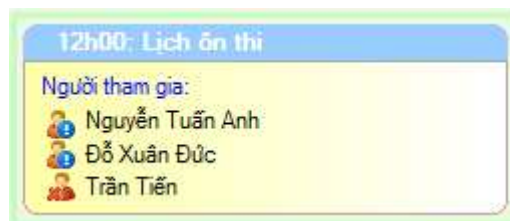
Hình 37. Màn hình duyệt tài nguyên

b) Hiển thị của một lịch trên dayView và weekview



Hình 38. Trình bày thông tin một lịch trên các view

c) Tip mô hiện thị thông tin lịch khi di chuột qua lịch



Hình 39. Tip của lịch (hiển thị khi di chuột qua lịch)

d) Giao diện tạo lịch mới

Calendar

Tập Công cụ Trợ giúp

Tạo lịch Lưu lại Tìm kiếm

Left Text Edit Box

Thông tin chung chi tiết

Thời gian bắt đầu 1/1/2011 10 : 00 Địa điểm Edit Box

Thời gian diễn ra 2:00

Người tham gia Tài nguyên Thêm

☐ Riêng tư ☒ Blocking

Table	Grid

Table	Grid

For Help, press F1

Hình 40. Cửa sổ tạo lịch mới

e) Tab chi tiết trên cửa sổ tạo lịch

☒ Báo thay đổi Lặp lại Không

Mức độ ưu tiên Bình thường

☐ Nhắc nhở Trước 15'

Đính kèm Edit Box

Nội dung

Mô tả về lịch

Hình 41. Tab chi tiết của cửa sổ tạo lịch