Trang bìa

Mô hình hóa tiến trình phát triển phần mềm với software prototyping và Mô hình phân tích động thái

Lời mở đầu

Trước đây, để khắc phục những vấn đề tồn tại trong cách tiếp cận hướng cấu trúc, người ta đã nghiên cứu một mô hình mới thích hợp cho việc phát triển phần mềm lớn và phức tạp, đó là mô hình hướng đối tượng. Cách tiếp cận hướng đối tượng đã ngày càng trở nên phổ biến. Trong các dự án phát triển hệ thống lớn, ngôn ngữ mô hình hóa hợp nhất ‐ UML đã được ưu tiên cho quá trình phân tích thiết kế hệ  thống. Ngày nay, nó được coi là một chuẩn quốc tế được tổ  chức tiêu chuẩn quốc tế ISO chấp nhận. Việc nắm vững các kiến thức cơ bản về  mô hình, quá trình mô hình hóa, các kỹ thuật xây dựng mô hình là những yêu cầu bắt buộc cho bất cứ ai muốn phân tích và thiết kế một hệ  thống lớn theo hướng đối tượng.

Trong bài báo cáo này, chúng ta sẽ đi vào tìm hiểu về ngôn ngữ UML và một số dạng mô hình UML.

Mục lục

I. Mô hình hóa tiến trình phát triển phần mềm

1. Tiến trình phát triển phần mềm - Mô hình software prototyping

Một tiến trình phát triển phần mềm là một tập của các hoạt động cần thiết để chuyển các yêu cầu người dùng thành một hệ thống phần mềm đáp ứng được các yêu cầu đặt ra.

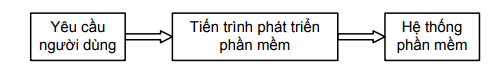


Figure 1: Tiến trình phát triển phần mềm cơ bản

Mô hình tiến trình phát triển phần mềm là sự mô tả tiến trình một cách đơn giản khi xem xét nó từ một cách nhìn cụ thể.

**Sofware Prototying** là một mô hình tiến trình phát triển phần mềm liên quan đến việc xây dựng một prototype của hệ thống, phần mềm hoặc một phần của hệ thống, phần mềm dùng để kiểm tra các chức năng, yêu cầu mà khách hành đưa ra và tính khả thi mà không đi sâu vào tính chính xác logic của hệ thống hay phần mềm đó.

Sofware Prototying đã trở thành một mô hình rất phổ biến và hữu dụng trong thiết kế hệ thống hay phần mềm bởi nó cho phép các nhà phát triển có thể hiểu và hình dung được các yêu cầu của hệ thống, phần mềm ngay từ giai đoạn rất sớm của quá trình phát triển.

Sofware Prototyping rất hữu dụng khi nhà phát triển không chắc chắn về các yêu cầu, thuật toán, tính hiệu quả hay tính khả thi của hệ thống, phần mềm.

2. Các giai đoạn thực hiện

Basic Requirement Identification

Developing the prototype

Review of the prototype

Revise and Enhance the prototype

Product

Giai đoạn 1: Xác địch mục tiêu, yêu cầu cơ bản Basic Requirement Identification

* Liên quan đến việc hiểu được những yêu cầu cơ bản nhất của phần mềm đặc biệt là UI (user interface). Những bước thiết kế dữ liệu bên trong và các khía cạnh như hiệu năng, bảo mật có thể hoàn tuàn được bỏ qua tại bước này.

Giai đoạn 2: Thiết kế prototype (Developing the Prototype)

* Việc thiết kế prototype được thực hiện ở giai đoạn này. Các tính năng căn bản nhất của phần mềm được thể hiện và UI được cài đặt.
* Để giảm thiểu chi phí prototype ban đầu, nhà phát triển có thể bỏ qua một số vấn đề như thời gian phản hồi, bộ nhớ sử dụng, …

Giai đoạn 3: Lấy ý kiến về prototype (Review of the prototype)

* Prototype đã được phát triển sẽ được sử dụng bởi một số khách hàng. Sau đó, các phản hồi được tập hợp lại và được sử dụng cho việc nâng cấp, cải tiến phần mềm.

Giai đoạn 4: Xem xét và cải tiến prototype (Revise and Enhance the prototype)

* Các phản hồi của khách hàng được xem xét, thống nhất và một prototype mới được phát triển. (Lặp lại giai đoạn 2)

Giai đoạn 5: Xuất sản phẩm Product

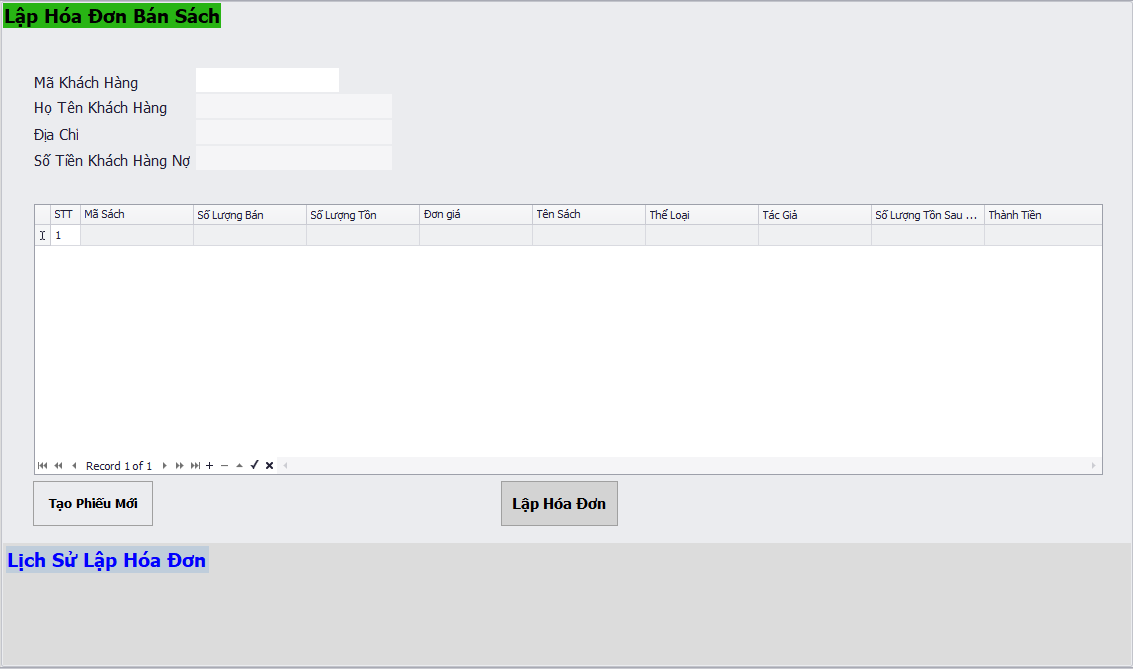
3. Ví dụ: Phần mềm quản lý nhà sách

Giai đoạn 1: Xác định yêu cầu cơ bản

* Các yêu cầu nghiệp vụ:
  + Lập phiếu nhập sách
  + Lập hóa đơn bán sách
  + Tra cứu sách
  + Thanh toán
  + Lập báo cáo tháng
  + Thay đổi quy định
* Đối tượng sử dụng phần mềm: nhân viên quản lý nhà sách.
* Phần mềm cần quản lý các tiến trình nhập, xuất sách.
* Phần mềm cần quản lý các đầu sách được nhập, xuất như tên sách, tác giả, thể loại, số lượng, …
* Khi nhân viên nhập, xuất sách cần tiến hành làm phiếu nhập và xuất.
* Phiếu tính tiền khi khách hàng mua sách tại nhà sách.
* Giao diện dễ sử dụng.
* …

**Ở các giai đoạn dưới đây chúng ta sẽ chỉ ví dụ về màn hình “Lập hóa đơn bán sách”**

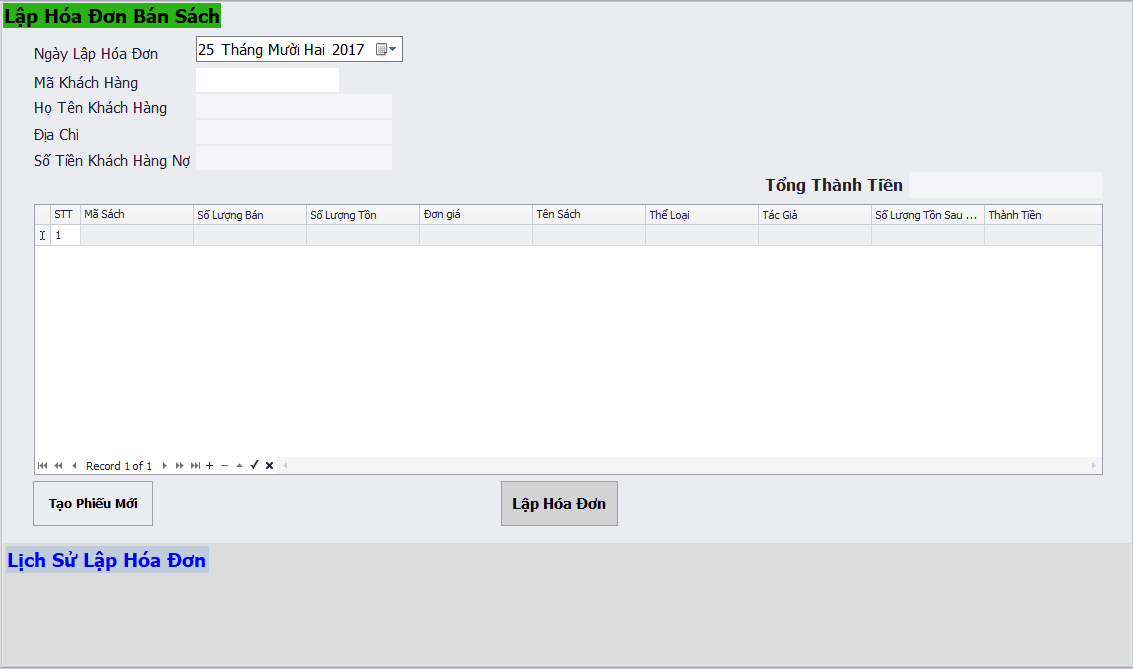
Giai đoạn 2: Thiết kế prototype

****

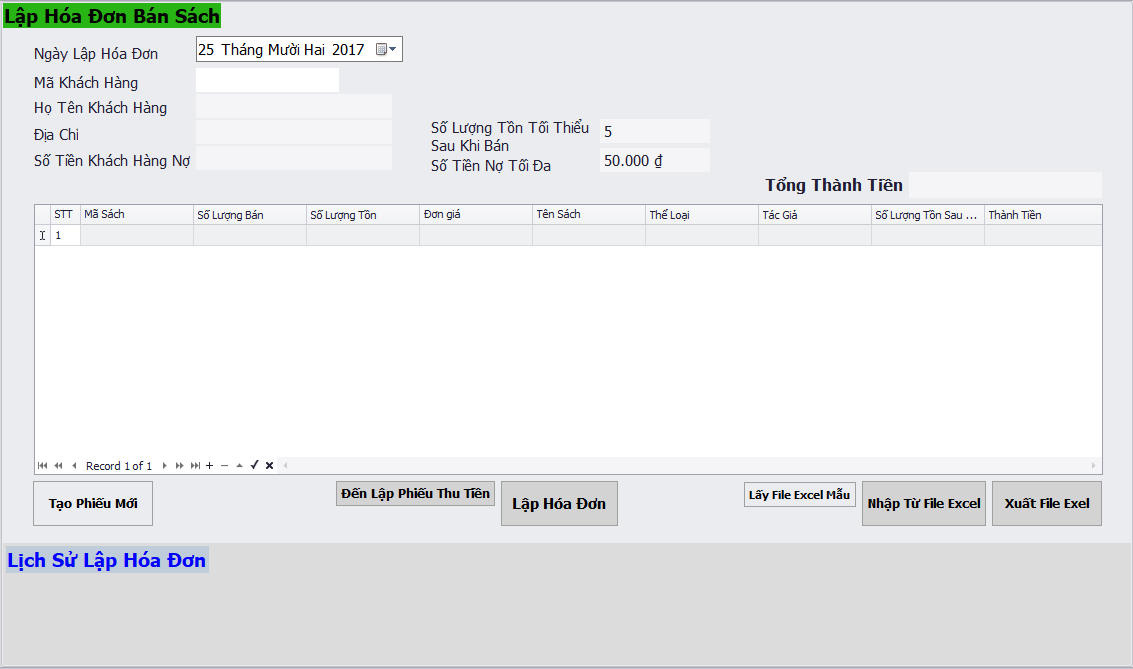
Giai đoạn 3: Lấy ý kiến về prototype

* Xác định các điểm bất hợp lý trong prototype:
* Chưa có thông tin về tổng thành tiền
* Thiếu ngày lập hóa đơn

Giai đoạn 4: Cải tiến prototype

****

Giai đoạn 5: Xuất sản phẩm

****

II. Ngôn ngữ UML và mô hình phân tích động thái:

1.UML:

**UML** (Tiếng Anh là Unified Modeling Language) là ngôn ngữ trực quan được dùng cho việc đặc tả, hình dung, xây dựng cho các hệ thống phần mềm. Cách tiếp cận theo mô hình UML giúp ích rất nhiều cho những người thiết kế cũng như những người sử dụng nó. UML tạo nên một cái nhìn bao quát và đầy đủ về hệ thống phần mềm dự định xây dựng.

UML là một tập các phần tử và một tập các quy tắc riêng. Các phần tử của UML là các đối tượng đồ họa như đường thẳng, hình chữ nhật và các quy tắc trong UML xác định các kết hợp giữa các phần tử này.

Trong UML, các đối tượng trao đổi với nhau bằng cách gửi các thông điệp cho nhau. Sự trao đổi này thể hiện sự tương tác giữa các đối tượng trong hệ thống. Mô hình động thái hệ thống được biểu diễn bằng các sơ đồ tương tác (Interaction Diagram)

- Sơ đồ trình tự (Sequence Diagram): mô tả sự trao đổi, tương tác của các đối tượng với nhau theo trình tự thời gian. Sơ đồ trình tự bao gồm các phần tử biểu diễn cho các đối tượng, các thông điệp được gửi và nhận trình tự theo thời gian để thực hiện các ca sử dụng của hệ thống.

- Sơ đồ trạng thái (State Diagram): mô tả các trạng thái, hành vi của các đối tượng. Sơ đồ trạng thái bao gồm những thông tin về những trạng thái khác nhau của các đối tượng, thể hiện các đối tượng chuyển từ trạng thái này sang trạng thái khác như thế nào, hành vi ứng xử của mỗi đối tượng khi có các sự kiện xảy ra để làm thay đổi trạng thái.

- Sơ đồ hoạt động (Activity Diagram): mô tả cách các đối tượng tương tác với nhau nhưng nhấn mạnh về công việc, xác định các hoạt động và thứ tự thực hiện những hoạt động đó.

- Sơ đồ cộng tác (Collaboration Diagram): mô tả sự tương tác của các đối tượng với nhau theo ngữ cảnh và không gian công việc.

Việc xây dựng các sơ đồ này nhằm mục đích mô hình hóa hoạt động của hệ thống, thực hiện việc gán trách nhiệm cho các đối tượng. Sơ đồ tương tác chỉ ra các tương tác, bao gồm tập đối tượng, quan hệ và các thông điệp trao đổi giữa chúng. Sơ đồ trình tự và sơ đồ tương tác chỉ ra từng bước của một luồng điều khiển cụ thể trong ca sử dụng. Từ sơ đồ tương tác, người thiết kế có thể phát hiện thêm các lớp, các thao tác cần thực hiện của mỗi lớp,... Do vậy, sơ đồ tương tác trở thành nền tảng cho các bước còn lại của quá trình phát triển phần mềm.

Không phải tất cả các hệ thống đều cần cả bốn sơ đồ tương tác trên mà nó còn tùy thuộc vào mức độ phức tạp của của hệ thống. Đối với những hệ thống tương đối đơn giản thì chỉ cần sơ đồ trình tự và sơ đồ trạng thái là đủ.

2. Các yếu tố thể hiện sự tương tác:

2.1. Các sự kiện và hành động của hệ thống:

Trong quá trình tương tác với hệ thống, các tác nhân gây ra các sự kiện làm cho hệ thống hoạt động và yêu cầu hệ thống phải thực hiện một số thao tác để đáp ứng các yêu cầu của những tác nhân đó. Các sự kiện phát sinh bởi các tác nhân có liên quan chặt chẽ với những hành động mà hệ thống cần thực hiện. Điều này suy ra là chúng ta phải xác định được các hoạt động của hệ thống thông qua các sự kiện mà các tác nhân gây ra.

Sự kiện là một hành động kích hoạt hệ thống để nó hoạt động, hoặc tác động lên hệ thống để nó hoạt động tiếp theo một cách nào đó. Nói cách khác, sự kiện là cái gì đó xảy ra và kết quả là nó có thể gây ra một số hoạt động sau đó của hệ thống. Ví dụ: Sau khi nhập vào hết các mặt hàng mà khách đã chọn mua, người bán hàng nhấn phím “Kết thúc”, thì hệ thống chuyển sang thực hiện chức năng thanh toán với khách mua hàng. Việc người bán hàng nhấn phím “Kết thúc” chính là sự kiện làm cho hệ thống chuyển sang trạng thái khác.

Các sự kiện có thể là độc lập hoặc có liên hệ với nhau. Ví dụ: Nhập thông tin về các mặt hàng và Thanh toán là hai sự kiện phụ thuộc, sự kiện Thanh toán phải xảy ra sau sự kiện nhập thông tin, còn sự kiện Trả tiền mặt và trả bằng séc là độc lập với nhau.

Những sự kiện độc lập có thể là những sự kiện đồng thời. Bởi vì những sự kiện này không phụ thuộc vào nhau nên có thể xảy ra trong cùng một thời điểm. Ví dụ: Hiển thị số tiền dư trả lại cho khách và Cập nhật các mặt hàng trong hệ thống bán hàng là hai sự kiện độc lập với nhau và có thể xảy ra đồng thời.

Các sự kiện cũng có thể chia thành hai loại: các sự kiện bên trong và các sự kiện bên ngoài.

* Sự kiện bên trong là sự kiện xảy ra ngay bên trong hệ thống, ở trong một đối tượng và được kích hoạt bởi đối tượng khác.
* Sự kiện bên ngoài là sự kiện được tạo ra ở bên ngoài phạm vi của hệ thống. Sự kiện vào của hệ thống là những sự kiện bên ngoài tác động vào hệ thống và do các tác nhân tạo ra.

Hoạt động của hệ thống là những thao tác mà hệ thống phải thực hiện để trả lời, đáp ứng cho những sự kiện vào. Một số hoạt động của hệ thống có thể tạo ra những sự kiện ra cho các tác nhân để thông báo những sự kiện tiếp theo của hệ thống có thể xảy ra, hoặc nhắc các tác nhân phải hành động như thế nào để có những thông tin mong muốn. Như vậy, các sự kiện vào sẽ kích hoạt hệ thống hoạt động và hệ thống hoạt động là để trả lời cho các sự kiện vào mà các tác nhân tạo ra.

Các sự kiện và hoạt động của hệ thống thường được sử dụng để mô tả “kịch bản” cho ca sử dụng.

2.2. Trao đổi thông điệp giữa các đối tượng:

Trong các sơ đồ tương tác, các đối tượng trao đổi với nhau bằng các thông điệp. Các đối tượng thường được gửi, nhận theo:

* Các giao thức trao đổi tin (Communication Protocol)
* Các lời gọi hàm: một đối tượng gọi một hàm của đối tượng khác để xử lý các yêu cầu.

3. Sơ đồ trình tự:

Theo yêu cầu của giai đoạn phân tích, chúng ta chỉ cần định nghĩa hệ thống như một hộp đen, trong đó hành vi của hệ thống thể hiện được những gì (What?) nó cần thực hiện và không cần thể hiện những cái đó thực hiện như thế nào (How?). Vì vậy, nhiệm vụ chính của chúng ta trong giai đoạn này là xác định và mô tả được các hoạt động của hệ thống theo yêu cầu của các tác nhân. Nghĩa là phải tìm được các sự kiện, các thao tác (sự tương tác) của các đối tượng trong từng ca sử dụng.

Sơ đồ trình tự giúp chúng ta thực hiện được những nhiệm vụ đó. Sơ đồ trình tự (sequence diagram) là sơ đồ tương tác theo trình tự thời gian của các giao tiếp bằng thông điệp giữa các đối tượng đang hoạt động trong hệ thống. Mỗi ca sử dụng có nhiều luồng dữ liệu. Mỗi sơ đồ trình tự biểu diễn một luồng dữ liệu.

3.1. Các thành phần của sơ đồ trình tự:

Sơ đồ trình tự bao gồm các phần tử biểu diễn đối tượng, thông điệp và thời gian. Sơ đồ trình tự được thể hiện theo hai trục:

* Trục dọc trên xuống chỉ thời gian xảy ra các sự kiện, hay sự truyền thông điệp, được biểu diễn bằng các đường gạch - gạch thẳng đứng bắt đầu từ đỉnh đến đáy của sơ đồ.
* Trục ngang từ trái qua phải là dãy các đối tượng tham gia vào việc trao đổi các thông điệp với nhau theo chiều ngang, có thể có cả các tác nhân.
* Đối tượng (object hoặc class) được biểu diễn bằng hình chữ nhật trong đó có tên đối tượng cụ thể và/hoặc tên lớp (hoặc tên lớp được gạch dưới biểu diễn cho một đối tượng bất kỳ của lớp đó).

C:\Users\2SMILE2\Desktop\fe845903-6f2b-4bb4-bd66-69e522af792b.png

* Đường đời đối tượng (Lifelines): biểu diễn bằng các đường gạch rời thẳng đứng bên dưới đối tượng



* Thông điệp (Message): biểu diễn bằng các đường mũi tên. Thông điệp được dùng để giao tiếp giữa các đối tượng và lớp.



* Xử lí bên trong đối tượng: biểu diễn bằng các đoạn hình chữ nhật rỗng nối với các đường đời đối tượng.



3.2. Các loại thông điệp trong sơ đồ trình tự:

* Thông điệp đồng bộ (Synchronous Message): Thông điệp đồng bộ cần có một request trước hành động tiếp theo.

C:\Users\2SMILE2\Desktop\eacbd1a0-1f31-4ded-9449-a998b4f85798.png

* Thông điệp không đồng bộ (Asynchronous Message): Thông điệp không đồng bộ không cần có một request trước hành động tiếp theo.

C:\Users\2SMILE2\Desktop\d4e8ce59-2466-42c0-9c8e-12eb3a64eed3.png

* Thông điệp chính mình (Self Message): Là thông điệp mà đối tượng gửi cho chính nó để thực hiện các hàm nội tại.

C:\Users\2SMILE2\Desktop\bcba9ebc-087a-439a-a0ab-0ba4e3a7ad23.png

* Thông điệp trả lời hoặc trả về (Reply or Return Message): Là thông điệp trả lời lại khi có request hoặc sau khi kiểm tra tính đúng đắn của một điều kiện nào đó. Ví dụ thông điệp loại này như tin nhắn trả về là success hoặc fail.

C:\Users\2SMILE2\Desktop\3bcad42c-e00e-4bbf-bcb4-348769354b62.png

* Thông điệp tạo mới (Create Message): Là thông điệp được trả về khi tạo mới một đối tượng.

C:\Users\2SMILE2\Desktop\9b45434e-fe70-4f26-a26e-842085449e50.png

* Thông điệp xóa (Delete Message): Là thông điệp được trả về khi xóa một đối tượng.

C:\Users\2SMILE2\Desktop\79309860-2782-4e92-a351-d8ab2021e988.png

Sơ đồ trình tự được đọc từ trên xuống dưới, từ trái sang phải. Thứ tự các đối tượng trong sơ đồ phải được sắp xếp sao cho đơn giản nhất có thể để dễ quan sát. Thời gian thực hiện một thông điệp của một đối tượng, hay còn gọi là hoạt động của đối tượng được biểu diễn bằng hình chữ nhật hẹp dọc theo trục thẳng đứng của đối tượng đó.

3.3. Các bước xây dựng sơ đồ trình tự:

- Xác định các tác nhân, các đối tượng tham gia vào ca sử dụng và vẽ chúng theo hàng ngang trên cùng theo đúng các ký hiệu.

- Xác định những thông điệp (lời gọi hàm) mà tác nhân cần trao đổi với một đối tượng nào đó, hoặc giữa các đối tượng tương tác với nhau theo trình tự thời gian và vẽ lần lượt các hoạt động đó từ trên xuống theo thứ tự thực hiện trong thực tế.

3.4. Ví dụ: Quản lý nhà sách

3.4.1. Sơ đồ trình tự mô tả ca sử dụng “Lập phiếu nhập sách”

Nhân viên nhà sách nhập vào các đầu sách sắp được nhập vào nhà sách. Khi nhập sách, nhân viên này cần thông tin và số lượng n của đầu sách được nhập. Những thông tin về đầu sách này được xác định thông qua mã sách id. Vậy đối tượng NhanVien gửi đến cho HeThong thông điệp nhập đầu sách với mã sách id và số lượng sách được nhập n nhapSach(id, n). Khi đã hoàn tất việc nhập tất cả các đầu sách thì cần phải báo cho hệ thống biết là đã nhập xong bằng cách nhấn nút HoanTatNhapSach. Điều này có nghĩa là thông báo cho hệ thống thông điệp capNhat().

Các hoạt động trên được mô tả bằng sơ đồ trình tự như sau:

capNhat()

nhapSach(id,n)

NhanVien

HeThong

3.4.2. Sơ đồ trình tự mô tả ca sử dụng “Lập hóa đơn bán sách”

Nhân viên nhà sách nhập vào các đầu sách sắp được bán. Khi thực hiện việc bán sách, nhân viên này cần thông tin và số lượng n của đầu sách được nhập. Những thông tin về đầu sách này được xác định thông qua mã sách id. Vậy đối tượng NhanVien gửi đến cho HeThong thông điệp bán đầu sách với mã sách id và số lượng sách được bán n banSach(id, n). Khi đã hoàn tất việc cung cấp thông tin tất cả các đầu sách được bán thì cần phải báo cho hệ thống biết là đã nhập xong thông tin bằng cách nhấn nút HoanTatLapHoaDon. Điều này có nghĩa là thông báo cho hệ thống thông điệp capNhat().

Các hoạt động trên được mô tả bằng sơ đồ trình tự như sau:

capNhat()

banSach(id,n)

NhanVien

HeThong

3.4.3. Sơ đồ trình tự mô tả ca sử dụng “Tra cứu sách”

Sau khi nhập các thông tin i1,i2,i3… của đầu sách cần tìm kiếm, nhân viên thông báo cho HeThong thông điệp timKiemSach(). HeThong nhận được thông điệp và hiển thị các đầu sách hienThiSach() phù hợp với thông tin tìm kiếm.

hienThiSach()

timKiemSach(i1,i2,i3,..)

NhanVien

HeThong

3.4.4. Sơ đồ trình tự mô tả ca sử dụng “Thanh toán”

Sau khi lập hóa đơn bán sách và nhân viên nhà sách bấm nút ThanhToan. Điều này có nghĩa là thông báo cho HeThong thông điệp batDauThanhToan(). HeThong sau đó hiển thị số tiền khách hàng phải trả hienThiTienPhaiTra(). Tiếp theo, NhanVien nhập vào số tiền (soTien) mà khách hàng đưa và nhấn nút hoanTatThanhToan. HeThong nhận được thông điệp thanhToan(soTien) và tính toán để hiển thị số tiền thừa phải trả lại khách hienThiTienThua().

Sơ đồ trình tự:

hienThiTienPhaiTra()

hienThiTienThua()

thanhToan(soTien)

batDauThanhToan()

NhanVien

HeThong

3.4.5. Sơ đồ trình tự mô tả ca sử dụng “Lập báo cáo tháng”

Nhân viên nhập tháng cần lập báo cáo và nhấn LapBaoCao. HeThong lúc này được thông báo thông điệp lapBaoCao(t) và trả về thông điệp baoCaoThang() cho nhân viên.

baoCaoThang()

lapBaoCao(t)

NhanVien

HeThong

3.4.6. Sơ đồ trình tự mô tả ca sử dụng “Thay đổi quy định”

Quản lý nhập các quy định cần thay đổi r1,r2, .. và nhấn ThayDoiQuyDinh. HeThong nhận được thông điệp thayDoiQuyDinh(r1,r2,..) và thực hiện capNhat() các quy định mới.

capNhat()

thayDoiQuyDinh(r1,r2,..)

QuanLy

HeThong

4. Sơ đồ trạng thái:

Sơ đồ trạng thái thể hiện chu kỳ hoạt động của đối tượng, các hệ thống con và của cả hệ thống, từ khi chúng được tạo ra cho đến khi kết thúc.

Sơ đồ trạng thái mô tả:

* Các trạng thái mà các đối tượng có thể có.
* Các sự kiện: các thông điệp nhận được, các lỗi có thể xuất hiện, điều kiện nào đó có thể trở thành đúng (true), khoảng thời gian đã qua,... tác động lên trạng thái để làm biến đổi.

Sơ đồ trạng thái là giải pháp tốt để mô hình hóa hành vi động của các lớp đối tượng. Trong một dự án, không nhất thiết phải tạo ra các sơ đồ trạng thái cho tất cả các lớp. Tuy nhiên, đối với những lớp có nhiều hành vi động, có nhiều trạng thái hoạt động khác nhau thì sơ đồ trạng thái là hữu ích, giúp chúng ta hiểu rõ hệ thống hơn.

4.1. Trạng thái và sự biến đổi trạng thai:

Mọi đối tượng trong hệ thống đều có chu kỳ sống và mỗi thời điểm đều có một trạng thái nào đó.

Trạng thái là một trong các điều kiện có thể để đối tượng tồn tại, là kết quả của một hoạt động trước đó của đối tượng. Trạng thái của đối tượng thường được mô tả trong hình chữ nhật góc tròn và được xác định bởi:

* Tên gọi trạng thái, thường bắt đầu bằng động từ.
* Biến trạng thái mô tả các giá trị hiện thời của trạng thái.
* Hoạt động là hành vi mà đối tượng sẽ thực hiện khi nó ở vào trạng thái đó.

4.2. Các thành phần của biểu đồ trạng thái:

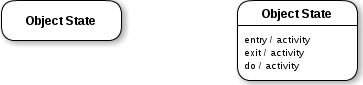
* Trạng thái bắt đầu (Initial State): C:\Users\2SMILE2\Desktop\4e28467b-de3f-414b-afed-4cd0cf209293.png
* Trạng thái kết thúc (Final State): C:\Users\2SMILE2\Desktop\fb72deda-1d56-427e-abb8-4cac505c3614.png

Trong biểu đồ, đường mũi tên chỉ ra sự biến đổi từ một trạng thái sang trạng thái khác.

* Sự kiện (Event) hoặc Chuyển đổi (Transition):

C:\Users\2SMILE2\Desktop\c5627259-a7d6-46e9-b9f1-c5623814a214.png

* Trạng thái đối tượng (State):



4.3. Xác định các trạng thái và các sự kiện:

Để xác định được các trạng thái và các sự kiện chúng ta cần trả lời cho các câu hỏi sau:

- Một đối tượng có thể ở những trạng thái nào? Liệt kê tất cả các trạng thái có thể có trong hệ thống của mỗi đối tượng.

- Những sự kiện nào có thể xuất hiện? Bởi vì sự kiện có thể làm biến đổi trạng thái, do vậy, từ các sự kiện có thể xác định được các trạng thái của đối tượng.

- Những trạng thái mới nào sẽ xuất hiện? Từ một trạng thái, đối tượng có thể chuyển sang trạng thái mới khi một số sự kiện xác định xuất hiện.

- Ở mỗi trạng thái, hoạt động của đối tượng là gì? - Sự tương tác giữa các đối tượng là gì? Sự tương tác giữa các đối tượng thường gắn chặt với các trạng thái của đối tượng.

- Những sự kiện, hay chuyển đổi trạng thái nào là không thể xảy ra? Một số sự kiện, hay trạng thái không thể chuyển đổi sang trạng thái khác được.

- Cái gì làm cho đối tượng được tạo ra? Đối tượng thường được tạo ra bởi một, hay một số sự kiện.

- Cái gì làm cho đối tượng bị hủy bỏ? Đối tượng thường được loại bỏ khi không còn cần thiết nữa?

4.4. Ví dụ: Quản lý nhà sách

4.4.1. Sơ đồ trạng thái mô tả “Lập phiếu nhập sách”

Một phiếu nhập sách được tạo lập

Nhập các đầu sách

Chưa nhập

Đã được nhập xong

Khi một phiếu nhập sách được tạo lập thì nó ở trạng thái chưa được nhập, sau khi được nhân viên nhập vào các đầu sách thì nó chuyển sang trạng thái đã được nhập xong.

4.4.2. Sơ đồ trạng thái mô tả “Lập hóa đơn bán sách”

Một hóa đơn bán sách được tạo lập

Nhập các đầu sách

Chưa nhập

Đã được nhập xong

Tương tự với “Lập phiếu nhập sách”, ta cũng có thể vẽ được sơ đồ:

Khi một hóa đơn bán sách được tạo lập thì nó ở trạng thái chưa được nhập, sau khi được nhân viên nhập vào các đầu sách được bán thì nó chuyển sang trạng thái đã được nhập xong.

4.4.3. Sơ đồ trạng thái mô tả “Tra cứu sách”

Tra cứu

Hiển thị sách phù hợp

Kết thúc tra cứu

Người dùng chuẩn bị tra cứu sách

Nhập các thông tin

Chờ để nhập các thông tin

Đã được nhập xong

Chờ tra cứu

Khi người dùng chuẩn bị tra cứu sách thì hệ thống đang chờ để được nhập các thông cần tìm kiếm. Sau khi các thông tin đã được nhập xong, việc tra cứu được thực hiện và hệ thống hiển thị ra cách đầu sách phù hợp với yêu cầu.

4.4.4. Sơ đồ trạng thái mô tả “Thanh toán”

Trả tiền

Hiển thị tiền thừa

Kết thúc thanh toán

Khách hàng chuẩn bị thanh toán

Hiển thị tiền phải trả

Chờ để thanh toán

Chờ thanh toán

Chờ kiểm duyệt tiền

Lấy tiền thừa

Khi khách hàng chuẩn bị thanh toán tiền thì khách hàng ở trạng thái chờ để thanh toán. Sau khi được hệ thống hiển thị tiền phải trả thì sơ đồ chuyển sang trạng thái chờ thanh toán. Tiếp đó, khách hàng trả tiền và chuyển sang trạng thái chờ kiểm duyệt tiền. Sau khi được nhân viên trả tiền thừa và chuyển sang trạng thái lấy tiền thừa thì việc thanh toán kết thúc.

4.4.5. Sơ đồ trạng thái mô tả “Lập báo cáo tháng”

Hiển thị báo cáo tháng

Một báo cáo tháng được tạo lập

Nhập tháng

Chưa nhập

Đã được nhập xong

Khi một báo cáo tháng được tạo lập, nó ở trang thái chưa nhập. Sau khi người dùng nhập vào tháng cần lập báo cáo, nó được nhập xong và hệ thống hiển thị báo cáo tháng.

4.4.6. Sơ đồ trạng thái mô tả “Thay đổi quy định”

Cập nhập các quy định

Chuẩn bị thay đổi quy định

Nhập các quy định

Chưa nhập

Đã được nhập

Khi người dùng chuẩn bị thay đổi quy định, hệ thống chờ để các quy định được nhập và sau khi các quy định được nhập xong, hệ thống thực hiện cập nhật các quy định đó.

III. Kết luận:

Việc xây dựng một bản thiết kế phần mềm không cần thiết phải xây dựng các sơ đồ như trên. Tuy nhiên, việc sử dụng UML giúp bạn tránh được nhiều khả năng thất bại trong quá trình phát triển phần mềm, có cái nhìn tường tận về những gì mà bạn đang làm, tận dụng khả năng sử dụng lại thành phần và vô số những tiện dụng khác.