

MÔN: ĐẶC TẢ PHẦN MỀM

Trần Văn Dũng, Bộ môn Khoa học Máy tính, biên soạn

PHẦN 1: BÀI GIẢNG

1. Khái niệm cơ bản đặc tả phần mềm: 6 tiết

- Đặc tả yêu cầu phần mềm
- Hướng dẫn viết đặc tả

2. Đặc tả thiết kế phần mềm: 15 tiết

- Đặc tả thiết kế
- Đặc tả các chức năng
- Đặc tả cơ sở dữ liệu
- Đặc tả các modules – Giao diện
- Đặc tả thiết kế hướng đối tượng

3. Mẫu thiết kế: 9 tiết Lý thuyết và 15 tiết Bài tập

Bài mở đầu và 23 mẫu thiết kế chính.

4. Bài tập lớn - Đặc tả phân tích và thiết kế ứng dụng nhỏ: 15 tiết

- Tuần 1: chọn Tên đề tài, mô tả yêu cầu.
- Tuần 2: Khảo sát, Phân tích yêu cầu
- Tuần 3: Thiết kế sơ đồ phân rã chức năng: đặc tả chức năng
- Tuần 4: Thiết kế sơ đồ quan hệ thực thể: đặc tả các thực thể, thuộc tính, quan hệ, ràng buộc.
- Tuần 5: Báo cáo cá nhân về kết quả trên
- Tuần 6: Cài đặt cơ sở dữ liệu vật lý
- Tuần 7: Thao tác SQL chèn dữ liệu mẫu vào CSDL
- Tuần 8: Đặc tả 2 modules và thời gian ôn thi: cài đặt hai modules để nghiệm thu khi thi
- Tuần 9 - 13: Cài đặt 23 mẫu thiết kế và xây dựng ít nhất ứng dụng 3 loại mẫu (mẫu tạo, mẫu cấu trúc và mẫu hành vi).
- Tuần 14-15: Nghiệm thu phần thực hành mẫu thiết kế

Kiểm tra điểm thành phần: 3.0

Đi học đầy đủ: 1.0

Nghiệm thu thực hành Design Pattern: 2.0

- Cài đặt 23 ứng dụng mẫu thiết kế: phân tích, giải thích ưu điểm khi cài đặt các mẫu thiết kế đó: 1.0
- Phát triển ứng dụng 3 mẫu thiết kế: mẫu tạo, mẫu cấu trúc và mẫu hành vi: 1.0

Thi - nghiệm thu bài tập lớn: 7.0

Đặc tả yêu cầu phần mềm

Đặc tả thiết kế

Đặc tả & cài đặt cơ sở dữ liệu

Đặc tả modules và cài đặt 2 form nhập dữ liệu

Kiểm tra chạy chương trình và nộp báo cáo khoảng 15 trang về các kết quả trên

Chương 1: Khái niệm cơ bản đặc tả phần mềm (6 tiết lý thuyết)

1. Vai trò đặc tả phần mềm
2. Đặc tả yêu cầu phần mềm
3. Các phương pháp đặc tả phần mềm

Chương 2: Đặc tả thiết kế phần mềm (15 tiết lý thuyết)

1. Đặc tả chức năng
2. Đặc tả cơ sở dữ liệu
3. Cài đặt cơ sở dữ liệu
4. Đặc tả module – Giao diện
5. Đặc tả thiết kế hướng đối tượng

Chương 3: Mẫu thiết kế (9 tiết lý thuyết – 15 tiết bài tập)

- Các mẫu tạo
- Các mẫu cấu trúc
- Các mẫu hành vi

Bài 1: Mở đầu (3 tiết)

- Kế hoạch môn học
- Mở đầu đặc tả
- Mẫu đặc tả yêu cầu phần mềm
- Hướng dẫn viết đặc tả
 - Đặt phòng khách sạn
 - Amazing Lunch

Bài 2: Đặc tả thiết kế (6 tiết)

- Cửa hàng mua bán trực tuyến
- Đặc tả sơ đồ phân rã chức năng:
 - Quản lý học tín chỉ
 - Đặc tả chức năng chính
 - Đặc tả chức năng cơ sở

Bài 3: Đặc tả thiết kế cơ sở dữ liệu (6 tiết)

- Thực thể, Thuộc tính, Quan hệ, Ràng buộc
- Mô hình quan hệ thực thể cho Hệ thống

- Bán vé máy bay
- Quản lý hàng điện tử
- Quản lý nhân sự

Bài 4: Đặc tả hình thức (6 tiết)

- Quan hệ
- Đặc tả modules – Đặc tả hình thức:
 - Ngăn xếp
 - Hệ thống hóa đơn
 - Hệ thống đa thang máy

Bài 5: Các mẫu cấu trúc (3 tiết lý thuyết – 4 tiết bài tập)

Adapter, Composite, Façade, Bridge, Flyweight, Proxy, Decorator

Bài 6: Các mẫu tạo (3 tiết lý thuyết – 4 tiết bài tập)

Singleton, Builder, Prototype, Factory method, Abstract Factory

Bài 7: Các mẫu hành vi (3 tiết lý thuyết – 7 tiết bài tập)

Iterator, Observer, Mediator, Chain of Responsibility, Template, Strategy, Command, State, Visitor

Bài 1: Đặc tả yêu cầu phần mềm

[TÊN PHẦN MỀM/DỰ ÁN]

ĐẶC TẢ YÊU CẦU PHẦN MỀM [v1.0]

Người lập:

Ngày lập:/...../.....

Người xem xét:

Ngày lập:/...../.....

Người duyệt:

Ngày duyệt:/...../.....

ĐD khách hàng duyệt:

Ngày duyệt:/...../.....

LỊCH SỬ THAY ĐỔI TÀI LIỆU

Ngày thay đổi	Phiên bản	Mô tả	Tác giả/Nhóm tác giả
<dd/mm/yy>	<Vx.y>	<Mô tả chi tiết Mục, bảng, sơ đồ nào thay đổi; nội dung thay đổi là gì>	<Họ tên người lập, nhóm lập>

MỤC LỤC

1.	GIỚI THIỆU CHUNG	6
1.1	Mục đích	6
1.2	Phạm vi	6
1.3	Các định nghĩa, thuật ngữ, từ viết tắt	6
1.4	Tài liệu tham khảo	6
2.	MÔ TẢ TỔNG QUAN HỆ THỐNG	7
2.1	Mô hình use case	7
2.2	Danh sách các tác nhân và mô tả	7
2.3	Danh sách Use case và mô tả	7
2.4	Các điều kiện phụ thuộc	7
3.	ĐẶC TẢ CÁC YÊU CẦU CHỨC NĂNG (FUNCTIONAL)	8
3.1	UC_001_Tên use case	8
3.1.1	Mô tả use case UC_001	8
3.1.2	Biểu đồ	8
3.2	UC_002_Tên use case	8
3.2.1	Mô tả use case UC_002	9
3.2.2	Biểu đồ	9
4.	CÁC THÔNG TIN HỖ TRỢ KHÁC	9

1. GIỚI THIỆU CHUNG

1.1 Mục đích

<Trình bày vai trò, mục đích của tài liệu **SRS**: Tài liệu mô tả một cách đầy đủ, toàn diện các yêu cầu của phần mềm – đó là các yêu cầu chức năng, phi chức năng, các ràng buộc về mặt thiết kế...>

<Ghi chú: tài liệu **SRS** mô tả các yêu cầu của phần mềm đối với toàn bộ hệ thống, hoặc đối với từng hệ thống con. Có nhiều kiểu cấu trúc cho tài liệu SRS. Cấu trúc giới thiệu trong tài liệu này là cấu trúc điển hình dùng cho các dự án áp dụng mô hình use-case (**use-case modeling**). Vì vậy, tài liệu sẽ trình bày các use case, mô tả cho các use case và các đặc tả bổ sung, cũng như các thông tin hỗ trợ khác>.

1.2 Phạm vi

<Mô tả ngắn gọn đặc điểm của phần mềm; lĩnh vực ứng dụng của phần mềm; phạm vi, đối tượng phục vụ của phần mềm; nhóm các hệ thống con, các mô hình Use-Case tương ứng...>

[Chỉ ra được tài liệu này dùng cho đối tượng nào?]

1.3 Các định nghĩa, thuật ngữ, từ viết tắt

<Mục này dành để giải thích cho các thuật ngữ và từ viết tắt dùng trong tài liệu, các định nghĩa sử dụng trong tài liệu. Có thể trình bày ngay trong mục này, cũng có thể tham chiếu tới một tài liệu riêng giải thích các thuật ngữ, từ viết tắt (gọi là Glossary) của dự án>.

STT	Thuật ngữ, từ viết tắt	Giải thích	Ghi chú

1.4 Tài liệu tham khảo

STT	Tên tài liệu	Ghi chú

2. MÔ TẢ TỔNG QUAN HỆ THỐNG

2.1 Mô hình use case

2.2 Danh sách các tác nhân và mô tả

<Liệt kê các tác nhân của hệ thống>.

Tác nhân	Mô tả tác nhân	Ghi chú

2.3 Danh sách Use case và mô tả

<Liệt kê các use case theo mô hình use case. Các use case tương ứng với các chức năng nào như đã mô tả trong tài liệu SRD. Phải mapping use case và chức năng tương ứng >.

ID	Tên Use case	Mô tả ngắn gọn Use case	Chức năng	Ghi chú
UC_001	Tên use case	Mô tả ngắn gọn Use case	Ứng với chức năng nào như đã mô tả trong SRD (mã số chức năng).	

Trong đó:

UC: Quy cách đánh số Use case

001, 002....: là số thứ tự của use case

2.4 Các điều kiện phụ thuộc

<Trình bày các điều kiện về mặt kỹ thuật cần thiết để phát triển phần mềm, ví dụ: cấu hình phần cứng, cấu hình mạng, các phần mềm khác có liên quan, các công cụ sử dụng để phát triển phần mềm. Các yêu cầu khác...>.

3. ĐẶC TẢ CÁC YÊU CẦU CHỨC NĂNG (FUNCTIONAL)

<Phần này mô tả một cách chi tiết từng yêu cầu cụ thể, cho phép các thành viên tham gia dự án căn cứ vào đó để xây dựng một phần mềm có chất lượng tốt nhất. Với cách tiếp cận theo mô hình UseCase (UC), các yêu cầu phần mềm được mô tả theo các UC và trong các đặc tả bổ sung>.

3.1 UC_001_Tên use case

Trong đó:

UC_001: là mã Use case

Tên use case: Tên Use case được mô tả ở mục 2.3

3.1.1 Mô tả use case UC_001

Use case: {Mã use case_Tên use case}	
Mục đích:	<Kết quả cần đạt được của Use case>.
Mô tả:	<Mô tả chi tiết use case, vai trò của Use case>
Tác nhân:	<Các tác nhân tác động đến Use case>
Điều kiện trước:	<Các điều kiện cần phải thực hiện trước khi thực hiện Use Case>.
Luồng sự kiện chính (Basic flows)	<Các luồng sự kiện chính, thành công của Use case theo trình tự thời gian>
Luồng sự kiện phụ (Alternative Flows):	<Các luồng sự kiện ngoại lệ, không thành công của Use case theo trình tự thời gian>
Điều kiện sau:	<Kết quả thu được sau khi thực hiện đúng & kết thúc UseCase>.

3.1.2 Biểu đồ

<Biểu đồ (diagram) chi tiết>

3.2 UC_002_Tên use case

Trong đó:

UC_002: là mã use case

Tên use case: Tên use case được mô tả ở mục 2.3

3.2.1 Mô tả use case UC_002

Use case: {Mã use case_Tên use case}	
Mục đích:	<Kết quả cần đạt được của Use case>
Mô tả:	<Mô tả chi tiết use case, vai trò của Use case>
Tác nhân:	<Các tác nhân tác động đến Use case>
Điều kiện trước:	<Các điều kiện cần phải thực hiện trước khi thực hiện Use Case>
Luồng sự kiện chính (Basic flows)	<Các luồng sự kiện chính, thành công của Use case theo trình tự thời gian>
Luồng sự kiện phụ (Alternative Flows):	<Các luồng sự kiện ngoại lệ, không thành công của Use case theo trình tự thời gian>
Điều kiện sau:	<Kết quả thu được sau khi thực hiện đúng & kết thúc UseCase>.

3.2.2 Biểu đồ

<Biểu đồ (diagram) chi tiết>.

4. CÁC THÔNG TIN HỖ TRỢ KHÁC

<Các thông tin hỗ trợ có vai trò làm cho tài liệu đặc tả yêu cầu phần mềm dễ sử dụng hơn, bao gồm: Mục lục, Index, Phụ lục kèm theo ...>.

Tài liệu tham khảo

Guidelines for Good Requirements Writing with Examples

Technical Report · May 2014
DOI: 10.13140/RG.2.1.5135.6108

Hướng dẫn viết đặc tả yêu cầu

Mục đích.

Mục đích của tài liệu này là cung cấp một số các qui tắc để viết đặc tả các yêu cầu. Khi kết hợp với các khía cạnh chất lượng của các yêu cầu, các hướng dẫn này sẽ tạo nên cơ sở để đạt được tài liệu yêu cầu tốt.

Các qui tắc

1. Sử dụng câu trực tiếp đơn giản
2. Sử dụng lượng từ vựng hạn chế
3. Xác định kiểu người dùng mà cần đến đặc tả yêu cầu
4. Tập trung vào khẳng định các kết quả đạt được
5. Xác định tiêu chí có thể kiểm tra được
6. Tránh nhập nhằng
7. Tránh tạo nên các yêu cầu đa mục tiêu
8. Tránh tạo nên các mệnh đề qui trách nhiệm không rõ ràng
9. Tránh thiết kế hệ thống
10. Tránh trộn lẫn giữa yêu cầu với thiết kế
11. Tránh trộn lẫn yêu cầu với kế hoạch
12. Tránh sự suy xét
13. Tránh chơi các yêu cầu nhập nhằng
14. Tránh dùng các thuật ngữ mơ hồ không xác định
15. Tránh lan man
16. Tránh thể hiện các khả năng
17. Tránh bày tỏ suy nghĩ mong muốn
18. Tránh đặc tả phủ định

Qui tắc 1

Sử dụng câu trực tiếp đơn giản

Mỗi yêu cầu cần là một câu tích cực đơn, càng ngắn càng tốt, nhưng không ngắn hơn.

Ví dụ

Hệ thống dẫn đường sẽ có khả năng nhận biết tốc độ máy bay.

Qui tắc 2

Sử dụng lượng từ vựng hạn chế

Viết sử dụng tập con các từ rõ nghĩa, tránh các thuật ngữ mà làm rối nghĩa người đọc không phải chuyên môn.

Ví dụ

- **Không tốt:** Máy bay sẽ có khả năng điều chỉnh lại chỗ ngồi cho hành khách thương gia/du lịch toàn cầu trước ít nhất nửa ngày (qui tắc FAA)
- **Tốt hơn:** Máy bay có khả năng thay đổi chỗ ngồi trên máy bay từ sử dụng kinh doanh sang sử dụng đặc quyền lễ hội trước ít nhất 12 tiếng.

Qui tắc 3

Xác định kiểu người dùng mà cần đến đặc tả yêu cầu

Mỗi yêu cầu cần được bắt đầu bởi tên của lớp người dùng

Ví dụ

Người **dẫn đường** cần có khả năng

Qui tắc 4

Tập trung vào khẳng định các kết quả đạt được

Mỗi yêu cầu cần định danh một kết quả mong muốn duy nhất. Một điều gì đó được cung cấp cho một lớp người dùng đã được định danh.

Ví dụ.

Người dẫn đường có khả năng nhìn thấy đám mây bão bằng rada

Qui tắc 5

Xác định tiêu chí có thể kiểm tra được

- Mỗi yêu cầu cần là kiểm tra được
- Bạn cần chỉ rõ kiểm tra có thể bằng việc bổ sung một câu đơn để kết nối với tiêu chí cụ thể đối với yêu cầu đó.
- Ở một bước sau này, tiêu chí cụ thể đó có thể được mở rộng thành tiêu chí chấp nhận
- Bạn không cần phải viết tiêu chí kiểm tra chấp nhận trong khi đang chuẩn bị các yêu cầu người dùng. Tuy nhiên, để kiểm tra thực tế, các tiêu chí kiểm tra được cần phải được xác định. Nếu nó không kiểm tra được, thì nó không là một yêu cầu.

Ví dụ

Tiêu chí đơn giản: Người dẫn đường có khả năng nhìn thấy mây bão bằng rada trước ít nhất 100 km

Tiêu chí chấp nhận: Máy bay bay ở vận tốc 800 km/giờ và độ cao 10000 m hướng đến mây bão được xác định bởi hệ thống vệ tinh; **mây bão được phát hiện trong vòng ít nhất 100 km**

Qui tắc 6

Tránh nhập nhằng

Một trong những vấn đề tinh tế và khó trong viết các yêu cầu

- Viết rõ ràng và tường minh
- Văn bản không hình thức, các lược đồ phác thảo, các cuộc trao đổi, gọi điện thoại có thể giúp loại bỏ sự không rõ nghĩa
- Sự nhập nhằng nguy hiểm có thể được gây ra bởi các từ “hoặc” chẳng hạn

Ví dụ

Một hệ thống con như vậy cũng có khả năng sinh ra tín hiệu cảnh báo/nhắc nhở bằng hình ảnh “hoặc” âm thanh để gây chú ý cho người lái phụ “hoặc” người dẫn đường.

Vấn đề

Hệ thống con nào? Tín hiệu có nhìn thấy hoặc nghe được không hoặc cả hai? Nó là cả hai cảnh báo và nhắc nhở, hay chỉ cảnh báo, hoặc chỉ nhắc nhở? ... gây chú ý cho lái phụ hoặc người dẫn đường hoặc cả hai?

Qui tắc 7

Tránh tạo nên các yêu cầu đa mục tiêu

Các yêu cầu mà chứa các liên từ như “và”, “hoặc”, “với”, “cũng” là sai lạc và nguy hiểm

Ví dụ

Đèn cảnh báo pin yếu sẽ sáng khi hiệu điện thế giảm xuống dưới 3.6 von, **và** không gian làm việc hiện thời **hoặc** dữ liệu đầu vào sẽ được lưu giữ lại.

Vấn đề

Ở đây có bao nhiêu yêu cầu, cái gì cần được lưu, và khi nào?

Qui tắc 8

Tránh tạo nên các mệnh đề qui trách nhiệm không rõ ràng

Các yêu cầu mà chứa các mệnh đề như “nếu”, “khi”, “nhưng”, “trừ khi”, “cho đến khi”, “mặc dù” là nguy hiểm và sai lạc

Ví dụ

- Các cửa hành khách phía trước sẽ được mở tự động khi máy bay dừng, **trừ khi** thang cửa sau được triển khai
- Chuông báo cháy sẽ reo khi khói được phát hiện, **trừ khi** chuông đang được kiểm tra hoặc các kỹ thuật viên đã ngắt chuông

Vấn đề

Ở đây có hai hay nhiều hơn các kịch bản trong cả hai ví dụ. Các mệnh đề kiểm trên đang thử bao quát hết các trường hợp ngoại lệ. Hãy kiểm soát các yêu cầu riêng biệt

Qui tắc 9

Tránh thiết kế hệ thống

- Các yêu cầu đặc tả lớp vỏ của thiết kế, và quá chi tiết sẽ bắt đầu thiết kế hệ thống, đặc biệt khi nó đi vào lĩnh vực ưu thích của chúng ta.
- Dấu hiệu nguy hiểm: đặt tên cho các thành phần, tài liệu, các thủ tục/đối tượng phần mềm, các trường dữ liệu.

Ví dụ

Ăng ten sẽ có khả năng nhận các tín hiệu FM, sử dụng dây lõi đồng với lớp bảo vệ nylon và lưới cao su làm cứng chống nước.

Vấn đề

Đặc tả thiết kế hơn nhu cầu thực tế thường làm tăng giá thành hệ thống bởi việc áp đặt các yêu cầu không cần thiết cho sự phát triển và chế tạo

Trong ví dụ này đã chỉ rõ Ăng ten cần có khả năng chống nước và va đập mạnh. Mục tiêu như vậy có thể đạt được sử dụng nguyên liệu tốt hơn đồng và rẻ hơn.

Qui tắc 10

Tránh trộn lẫn giữa yêu cầu với thiết kế

- Các yêu cầu của người dùng tạo nên mô hình hoàn chỉnh những gì mà người sử dụng mong muốn
- Đặc tả hệ thống tạo nên mô hình hoàn chỉnh của hệ thống được đề xuất
- Sự lẫn lộn có thể xảy ra khi trộn lẫn các yêu cầu người dùng với các đặc tả hệ thống, các yếu tố thiết kế, các kịch bản kiểm thử, các hướng dẫn phát triển và các chỉ dẫn cài đặt.

Ví dụ

Người dùng có thể nhìn thấy số kênh đang chọn hiện thời mà được hiện lên ở dạng 14 pt Swiss trên bảng LCD được kiểm tra theo chuẩn Liên đoàn 567-89 và gắn chặt với vòng đệm cao su chống va đập.

Vấn đề

Trộn lẫn các yêu cầu và các ràng buộc thể hiện. Lựa chọn bảng LCD có thể để sau khi hệ thống được thiết kế. Tính chất chắc chắn của LCD có thể được xem xét như một ràng buộc.

Qui tắc 11

Tránh trộn lẫn yêu cầu với kế hoạch

- Trộn lẫn yêu cầu với kế hoạch và lập lịch là nguồn gốc của rắc rối.
- Kế hoạch là cần thiết nhưng không thuộc yêu cầu.
- Trộn lẫn cả hai làm tăng luồng công việc yêu cầu thông qua các phiên bản và các cuộc gặp chỉnh sửa thừa.
- Thông thường chúng được cập nhật xuyên suốt dự án trong khi các yêu cầu cần được ổn định.
- Các dấu hiệu nguy hiểm: các ngày tháng, các pha dự án và các hoạt động phát triển.

Ví dụ

Các kiểu thể hiện kênh – LCD, LED hoặc TFT – sẽ được chọn khoảng 15 tháng ba và bảng mẫu đầu tiên sẽ sẵn sàng để kiểm thử vào đầu giai đoạn 3.

Vấn đề

Mặc dù câu trên được thể hiện như một yêu cầu, nhưng nó không phải là yêu cầu.

Qui tắc 12

Tránh sự suy xét

- Các yêu cầu là một phần của hợp đồng giữa khách hàng và người phát triển, với người sử dụng như là bên thứ ba quan tâm.
- Ở đây không có không gian cho “danh sách mong muốn” – các thuật ngữ nói chung về những việc mà một người nào đó mong muốn.
- Chữ ký nguy hiểm bao gồm tính không rõ ràng về phát ngôn những từ chung chung: thông thường, nói chung, thường là, một cách bình thường.

Ví dụ

Người sử dụng thông thường yêu cầu dấu hiệu sớm về sự xâm nhập vào hệ thống.

Vấn đề

Câu này không chứa bất cứ ràng buộc hoặc yêu cầu nào trên hệ thống, mặc dù để cho người phát triển hệ thống quyết định bổ sung đặc tính này hoặc bỏ qua nó, vì người đó có thể nghĩ về kịch bản mà ở đó điều này không là trường hợp bình thường.

Qui tắc 13

Tránh chơi các yêu cầu nhập nhằng

- Việc sử dụng các từ “hoặc” và “trừ khi” trong các yêu cầu, cho phép các nhóm người đọc khác nhau hiểu các việc khác nhau từ cùng một câu chữ.
- Người quản trị phát triển có thể trì hoãn cho đến muộn nhất có thể những điều mà khách hàng yêu cầu là mong muốn có.
- Khách hàng có thể hy vọng làm cho người cung cấp nghĩ sai để đề xuất hoàn thành công việc với giá thấp hơn.

Ví dụ

Các nhân viên vận hành sẽ có khả năng sao lưu đĩa bất kỳ vào một ổ đĩa di động tốc độ cao hoặc băng từ.

Vấn đề

Điều trên có thể được hiểu theo nghĩa là “một ổ đĩa tốc độ cao” hoặc “một băng từ tốc độ cao” mà được giả thiết trước là người sử dụng muốn, nó cũng có thể có nghĩa là “một ổ đĩa tốc độ cao” hoặc “bất cứ loại băng từ nào”, mà có thể là rẻ hơn cho người phát triển để cung cấp nó, nếu các mong muốn của người sử dụng có thể bỏ qua.

Qui tắc 14

Tránh dùng các thuật ngữ mơ hồ không xác định

- Nhiều từ được sử dụng không hình thức để chỉ ra chất lượng mong muốn mà là quá mập mờ để sử dụng trong các yêu cầu.
- Các thuật ngữ mập mờ bao gồm: **thân thiện người sử dụng, linh hoạt, xấp xỉ, như có thể, hiệu quả, nâng cấp, hiệu năng cao, hiện đại.**
- Các yêu cầu sử dụng các thuật ngữ trên là không được kiểm chứng vì không có kiểm tra xác định để chỉ ra hệ thống như vậy có tính chất được chỉ ra không.

Ví dụ

- Hộp thoại Print là linh hoạt và thân thiện người sử dụng
- Đèn chỉ báo trạng thái OK sẽ sáng sớm nhất có thể sau khi tự kiểm tra hệ thống hoàn tất.

Vấn đề

Linh hoạt là gì? Và thân thiện người sử dụng là gì? Sớm nhất có thể là khi nào?

Qui tắc 15

Tránh lan man

Các câu lan man dài dòng sẽ dẫn đến mập mờ và lỗi.

Ví dụ

Được cung cấp rằng các tín hiệu đầu vào được chỉ định từ các thiết bị chuyên dụng là nhận được theo thứ tự đúng ở đó hệ thống có khả năng phân biệt các người chỉ định, tín hiệu đầu ra sẽ tuân theo với khung yêu cầu ở mục 3.1.5 để chỉ ra trạng thái đầu vào mong muốn.

Vấn đề

Nhiều thông tin không cần thiết trong ví dụ này, mà chỉ tạo ra sự mập mờ và tất yếu là hiểu sai lệch mục đích chính.

Qui tắc 16

Tránh thể hiện các khả năng

- Các khả năng được chỉ ra với các thuật ngữ như: có thể, có lẽ, có khả năng, ...
- Nếu các nhà phát triển chỉ làm những gì họ phải làm, họ sẽ luôn bỏ qua các điều mà người sử dụng có thể có khả năng yêu cầu

Ví dụ

Hệ thống tiếp tân có thể sẽ cần phải quan tâm để nhận tín hiệu bên trong tòa nhà cốt thép

Vấn đề

Quyết định thiết kế dựa trên khẳng định trên có thể dẫn đến hệ thống sai. Như là có khả năng gây ra nguyên nhân để nhà phát triển bỏ qua nó hoàn toàn. Sẽ có ít công việc và nỗ lực hơn, nhưng dẫn đến hệ thống sai.

Qui tắc 17

Tránh bày tỏ suy nghĩ mong muốn

- Công nghệ là hoạt động thực tế
- Không có hệ thống hoặc thành phần nào hoàn hảo
- Suy nghĩ mong muốn có nghĩa là yêu cầu điều không có thể
- Các người phát triển sẽ bác bỏ các yêu cầu mong muốn một cách đúng đắn
- Thuật ngữ mong muốn bao gồm: 100% tin cậy, an toàn, kiểm soát mọi lỗi không lường trước, làm hài lòng mọi người, chạy trên mọi nền tảng, không bao giờ chết, có thể nâng cao chất lượng cho mọi tình huống lỗi

Ví dụ

- Hộp số sẽ là an toàn 100% trong thao tác bình thường
- Mạng sẽ kiểm soát mọi lỗi không lường trước mà không bị sập

Vấn đề

An toàn 100% là gì? Làm thế nào để kiểm soát các lỗi không lường trước?

Qui tắc 18

Tránh đặc tả phủ định

- Sử dụng đặc tả phủ định hạn chế, chỉ để nhấn mạnh
- Không sử dụng đặc tả phủ định cho các yêu cầu mà có thể được khẳng định ở dạng chắc chắn.

Ví dụ

Người sử dụng sẽ không bị ngăn cản xóa dữ liệu mà họ nhập vào

Vấn đề

Đặc tả phủ định.

- Hệ thống sẽ cho phép người sử dụng có khả năng xóa dữ liệu mà họ nhập vào

Ví dụ: Đặc tả yêu cầu ứng dụng Tìm quán ăn: Amazing Lunch Indicator

Sarah Geagea 881024-4940

Sheng Zhang 850820-4735

Niclas Sahlin 880314-5658

Faegheh Hasibi 870625-5166

Farhan Hameed 851007-9695

Elmira Rafiyan 840724-5383

Magnus Ekberg 851022-1933

1. Mở đầu

Mục này cho mô tả phạm vi và tổng quan về mọi vấn đề được nằm trong tài liệu đặc tả yêu cầu.

1.1 Mục tiêu

Mục tiêu là cho mô tả chi tiết các yêu cầu của ứng dụng “Amazing Lunch Indicator” (ALI). Nó sẽ mô tả mục đích và khai báo yêu cầu hoàn chỉnh để phát triển hệ thống. Nó để giải thích các ràng buộc hệ thống, giao diện và tương tác với các ứng dụng bên ngoài. Tài liệu này dành cho khách hàng để nhất trí về yêu cầu và dùng để tham chiếu để phát triển phiên bản đầu tiên của hệ thống dành cho đội ngũ phát triển.

1.2 Phạm vi

Ứng dụng ALI là ứng dụng trên điện thoại di động dựa trên GPS, mà nó giúp người sử dụng tìm nhà hàng gần nhất dựa trên vị trí người dùng và các tiêu chí khác như giá cả, kiểu nhà hàng, món ăn và nhiều thứ khác nữa. Ứng dụng cần là tự do tải về từ một kho ứng dụng trên điện thoại di động nào đó hoặc các dịch vụ tương tự.

Các chủ nhà hàng có thể cung cấp thông tin nhà hàng của họ sử dụng cổng web (web-portal). Ứng dụng này sẽ hoạt động như cơ sở để hiển thị kết quả tìm kiếm cho người sử dụng. Người quản trị sẽ dùng web-portal để quản trị hệ thống và đảm bảo thông tin chính xác. Người quản trị có thể, chẳng hạn, kiểm tra thông tin người chủ khách sạn và quản trị thông tin người sử dụng.

Ngoài ra, phần mềm cần cả kết nối Internet và GPS để lấy và hiển thị kết quả. Mọi thông tin hệ thống được bảo trì trong cơ sở dữ liệu, mà được đặt trên một web server nào đó. Phần mềm cũng tương tác với phần mềm GPS-Navigator mà được yêu cầu đã cài đặt sẵn trên điện thoại di động của người sử dụng. Bằng việc sử dụng GPS-Navigator, người sử dụng có thể xem các nhà hàng mong muốn trên bản đồ và có thể được định hướng đi đến đó. Ứng dụng cũng có khả năng hiển thị cả thông tin tổng quan và chi tiết về các nhà hàng.

1.3 Các định nghĩa, thuật ngữ và từ viết tắt

Thuật ngữ	Định nghĩa
Người sử dụng	Một người nào đó tương tác với ứng dụng trên điện thoại di động
Người quản trị	Quản trị hệ thống mà được cho phép để quản trị và kiểm soát hệ thống
Chủ nhà hàng	Một ai đó mà có nhà hàng và muốn nhà hàng này là một phần của ứng dụng
Web-Portal	Ứng dụng Web mà cung cấp các tiện ích cho chủ nhà hàng và quản trị
GPS	Global Positioning System – Hệ thống định vị toàn cầu
GPS-Navigator	Phần mềm được cài đặt trên điện thoại di động mà cung cấp kết nối GPS và dữ liệu, chỉ ra các vị trí trên bản đồ và tìm đường đi từ vị trí hiện tại đến các đích xác định
Kho ứng dụng	Một ứng dụng được cài đặt trên điện thoại di động mà trợ giúp người sử dụng tìm các ứng dụng mới tương thích với nền tảng của điện thoại di động và tải chúng về từ Internet.

DESC	Description – mô tả
RAT	Rational – Quan hệ
DEP	Dependency – Phụ thuộc

1.4 Tài liệu tham khảo

1. IEEE Software Engineering Standards Committee. “IEEE Std 830-1998, IEEE Recommended Practice for Software Specifications”, October 20, 1998.

1.5 Tổng quan

Phần còn lại của tài liệu này gồm 3 chương và phụ lục. Chương 2 cung cấp tổng quan về chức năng hệ thống và tương tác của hệ thống với các hệ thống khác. Chương này cung cấp ra các kiểu người sử dụng khác nhau và tương tác của họ với hệ thống. Tiếp theo chương cũng nói đến các ràng buộc hệ thống và các giả thiết về sản phẩm.

Chương 3 cung cấp chi tiết các đặc tả yêu cầu và mô tả các giao diện hệ thống khác nhau. Các kỹ thuật ứng dụng khác nhau được sử dụng để đặc tả yêu cầu chính xác hơn cho các bạn đọc khác nhau.

Chương 4 và phụ lục đề cập đến sự ưu tiên của các yêu cầu.

2. Mô tả tổng quan

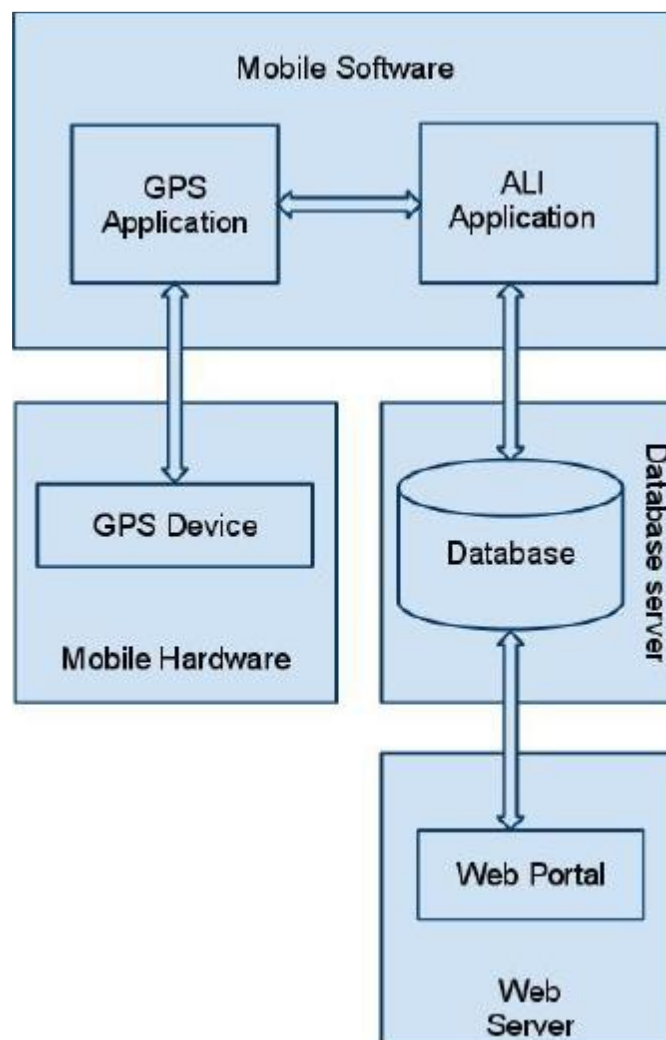
Mục này cung cấp tổng quan về toàn bộ hệ thống. Hệ thống sẽ được giải thích trong ngữ cảnh của nó để chỉ ra hệ thống tương tác như thế nào với các hệ thống khác và đưa ra các chức năng cơ bản của nó. Nó cũng sẽ mô tả các kiểu người mà sẽ sử dụng hệ thống và chức năng nào sẵn sàng cho mỗi kiểu người đó. Cuối cùng các ràng buộc và các giả thiết cho hệ thống sẽ được trình bày.

2.1 Mô tả sản phẩm

Hệ thống sẽ gồm hai phần: một ứng dụng di động và một web-portal. Ứng dụng di động sẽ được sử dụng để tìm kiếm nhà hàng và xem thông tin về chúng trong khi web portal sẽ được sử dụng để quản trị thông tin về các nhà hàng và hệ thống nói chung.

Ứng dụng di động sẽ cần trao đổi thông tin với ứng dụng GPS bên trong điện thoại di động, mà đến lượt nó sẽ trao đổi với thiết bị GPS vật lý để tìm vị trí người sử dụng. GPS sẽ cung cấp ứng dụng di động với vị trí của cả người sử dụng và các nhà hàng và khoảng cách giữa chúng, cũng như nó sẽ cung cấp bản đồ và chức năng hiển thị dữ liệu ứng dụng trên bản đồ. Chức năng này được cung cấp bởi GPS sẽ được nhúng vào trong ứng dụng để cho người sử dụng có thể sử dụng các chức năng này một cách thuận tiện.

Vì đây là sản phẩm lấy dữ liệu làm trọng tâm, nó sẽ cần nơi nào đó để lưu giữ dữ liệu. Do đó, cơ sở dữ liệu sẽ được sử dụng. Cả ứng dụng di động và web portal sẽ trao đổi với cơ sở dữ liệu, tuy nhiên theo các cách khác nhau. Ứng dụng di động sẽ sử dụng cơ sở dữ liệu để lấy dữ liệu, còn web portal sẽ bổ sung và chỉnh sửa dữ liệu. Mọi trao đổi với cơ sở dữ liệu đều đi qua Internet.



2.2 Các chức năng của sản phẩm

Với ứng dụng di động, người sử dụng có khả năng tìm kiếm các nhà hàng. Kết quả sẽ dựa trên các tiêu chí người sử dụng nhập vào. Có một số tiêu chí tìm kiếm và nó có thể được người quản trị của hệ thống quản lý các lựa chọn của các tiêu chí này.

Kết quả tìm kiếm sẽ được trình bày dưới dạng danh sách hay dạng bản đồ, tùy thuộc vào các tiêu chí được đưa vào để tìm kiếm. Dạng danh sách sẽ có một danh sách các đồ vật cho mỗi nhà hàng khớp với tiêu chí tìm kiếm và đưa ra một phần nhỏ thông tin nhà hàng mà người sử dụng có thể định danh nhà hàng. Dạng bản đồ sẽ chỉ ra vị trí nhà hàng như cái ghim trên bản đồ, cũng như vị trí của người sử dụng. Trong cả hai dạng thể hiện người sử dụng có khả năng chọn nhà hàng như đích đến hoặc nhận thông tin đến đó bằng cách nào, hoặc xem các thông tin của một nhà hàng cụ thể.

Web portal sẽ cung cấp chức năng để quản trị hệ thống và thông tin của nhà hàng. Nó cũng cung cấp thông tin về hệ thống, chẳng hạn chỉ ra khi có các thay đổi mới.

2.3 Các đặc tính của người sử dụng

Ở đây có ba loại người dùng mà tương tác với hệ thống: người sử dụng ứng dụng di động, chủ nhà hàng và người quản trị. Mỗi một trong ba loại người sử dụng trên sử dụng hệ thống theo cách khác nhau, như vậy mỗi loại trong họ có những yêu cầu của mình khác nhau.

Người sử dụng điện thoại di động có thể chỉ dùng điện thoại để tìm nhà hàng. Điều này có nghĩa là người sử dụng có khả năng tìm kiếm nhà hàng, chọn nhà hàng từ tìm kiếm đó và sau đó tìm đường đi đến đó. Để người sử dụng nhận kết quả tìm kiếm liên quan, ở đây có nhiều tiêu chí người sử dụng có thể có thể đặc tả và mọi kết quả khớp với tất cả tiêu chí đó.

Người chủ nhà hàng sẽ không dùng điện thoại di động, nhưng sẽ dùng web portal. Họ sẽ quản trị thông tin về nhà hàng của họ, chẳng hạn mô tả nhà hàng, thông tin liên lạc và các thực đơn.

Người quản trị cũng chỉ tương tác với web portal. Họ quản trị toàn bộ hệ thống sao cho không có thông tin không chính xác ở đó. Người quản trị có thể quản lý thông tin cho mỗi nhà hàng cũng như các lựa chọn cho cả người sử dụng ứng dụng di động và các chủ nhà hàng.

2.4 Các ràng buộc

Ứng dụng di động được ràng buộc bởi giao diện hệ thống cho hệ thống chỉ đường GPS ở bên trong điện thoại di động. Vì ở đây có nhiều hệ thống và nhiều nhà cung cấp GPS, giao diện sẽ không như nhau đối với mỗi một trong số đó. Cũng như vậy, ở đây có thể có sự khác biệt trong các đặc trưng định hướng mà mỗi hệ thống cung cấp.

Kết nối Internet cũng là ràng buộc cho ứng dụng. Vì ứng dụng lấy dữ liệu từ cơ sở dữ liệu qua Internet, nên có tính quyết định là phải có kết nối Internet cho ứng dụng để vận hành.

Cả hai web portal và ứng dụng di động sẽ được ràng buộc bởi khả năng của cơ sở dữ liệu. Vì cơ sở dữ liệu là chia sẻ giữa cả hai ứng dụng nó có thể buộc phải xếp hàng các yêu cầu đến và do đó tăng thời gian để nó lấy được dữ liệu.

2.5 Các giả thiết và các yêu cầu

Một giả thiết về sản phẩm là nó luôn được sử dụng trên điện thoại di động mà có hiệu năng chấp nhận được. Nếu điện thoại không có đủ tài nguyên phần cứng sẵn sàng cho ứng dụng, mà người sử dụng có thể được cấp với các ứng dụng khác, thì ở đây có kịch bản mà ứng dụng không làm việc được như mong muốn.

Một giả thiết khác là các thành phần GPS trong các điện thoại làm việc theo cách như nhau. Nếu điện thoại có các giao diện khác nhau đến GPS, thì ứng dụng cần được chỉnh cho mỗi giao diện và nó có nghĩa là việc tích hợp với GPS có thể có các yêu cầu khác nhau so với những gì có trong đặc tả này.

3. Các yêu cầu chuyên biệt

Mục này chứa mọi yêu cầu chức năng và chất lượng của hệ thống. Nó cung cấp mô tả chi tiết hệ thống và mọi đặc tính của nó.

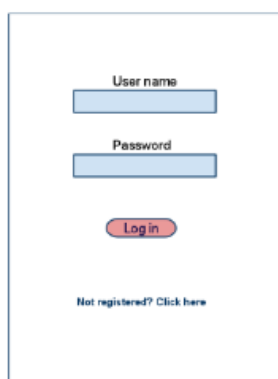
3.1 Các yêu cầu giao diện bên ngoài

Mục này cung cấp mô tả chi tiết về mọi đầu vào và đầu ra từ hệ thống. Nó cũng mô tả phần cứng, phần mềm và giao diện trao đổi thông tin và cung cấp các nguyên bản của giao diện người sử dụng.

3.1.1 Giao diện người sử dụng

Người sử dụng lần đầu của ứng dụng di động cần phải nhìn thấy trang đăng nhập khi họ mở ứng dụng. Nếu người sử dụng chưa đăng ký, họ cần phải làm điều đó trên trang đăng nhập.

Nếu người sử dụng không phải sử dụng lần đầu, thì họ cần được nhìn thấy trực tiếp trang tìm kiếm khi ứng dụng được mở ra. Ở đây người sử dụng chọn kiểu tìm kiếm mà họ muốn tạo ra.



A login screen with a light blue background. It features two input fields: 'User name' and 'Password'. Below these is a red 'Log in' button. At the bottom, there is a link that says 'Not registered? Click here'.

Trang Đăng nhập



A search screen with a light blue background. It has a 'Free-text search' input field at the top. Below it, there is a section 'Or search by:' with four checkboxes: 'Destination', 'Price', 'Restaurant type', and 'Dish'. Each checkbox has associated input fields for 'Min' and 'Max' values. For 'Destination', the units are '0 km' and '10 km'. There is a red 'Search' button at the bottom. A small circular icon with the letter 'P' is in the bottom right corner.

Trang Tìm kiếm



A user profile screen with a light blue background. It has a 'Back' button in the top left. The 'User Name' field shows 'nama@example.com' with an 'Edit' button. Below it, a phone number '070 - 400 00 00' is shown with an 'Edit' button. A 'Change password' button is below the phone number. At the bottom, there is a 'Change language' section with four buttons: 'Swedish', 'French', 'English', and 'Spanish'.

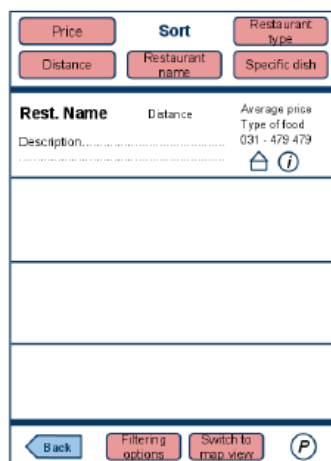
Trang Hồ sơ

Mỗi người sử dụng cần phải có trang hồ sơ ở đó họ có thể sửa địa chỉ email, số phone và mật khẩu của họ. Cũng như vậy, người sử dụng cần phải thiết lập cho ứng dụng di động ngôn ngữ mà họ mong muốn. Biểu tượng “P” chỉ ra chỗ người sử dụng có thể kích để điều hướng đến trang hồ sơ của họ.

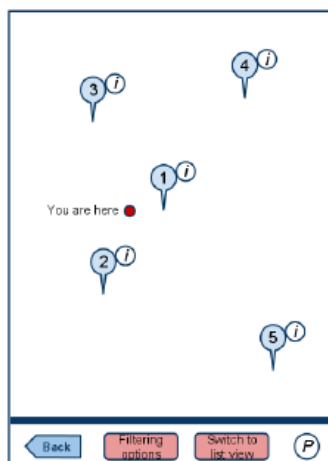
Xem dạng danh sách được chỉ ra trên Hình sau. Khi người sử dụng tìm theo giá, xem dạng này cần phải là dạng danh sách mặc định. Tiêu đề sắp xếp cho phép người sử dụng sắp xếp kết quả theo giá, tên nhà hàng, khoảng cách, kiểu nhà hàng và món ăn chuyên biệt. Mỗi mục kết quả bao gồm thông tin về nhà hàng và link đến trang web của nhà hàng đó và đường link thông tin mà cung cấp mô tả chi tiết hơn của nhà hàng. Cũng có lựa chọn lọc, ở đó người sử dụng có thể chọn để lọc kết quả bằng cách tăng hay giảm giá hay phạm vi khoảng cách.

Trên dạng xem bản đồ, mỗi khách sạn được biểu diễn bởi dạng một cái ghim. Cạnh mỗi ghim là link thông tin mà cung cấp mô tả chi tiết hơn của nhà hàng. Cũng lựa chọn lọc như vậy như trong dạng xem danh sách, được đưa vào dạng xem bản đồ.

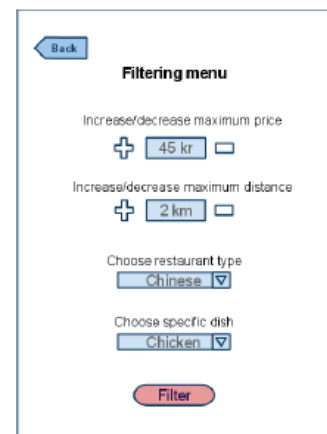
Người chủ nhà hàng và quản trị tương tác với hệ thống qua web portal. Chủ nhà hàng cần phải đăng ký trên web portal để đăng nhập và quản trị thông tin nhà hàng. Người quản trị cần đăng nhập vào web portal ở đó họ có thể quản trị hệ thống bằng cách hiệu chỉnh thông tin nhà hàng và người sử dụng.



Xem dạng danh sách



Xem dạng bản đồ



Lọc Menu

Web-portal

First time here? Register and make your restaurant visible to everyone!

Register now!

Log in as:

☐ Restaurant owner

☐ Administrator

User name

Password

Log in

Web portal

This section includes the requirements that specify all the fundamental actions of the software system.

3.2 Các chức năng

Mục này bao gồm các yêu cầu mag đặc tả mọi hành động chức năng của hệ thống phần mềm.

3.2.1 Lớp người sử dụng 1

3.2.1.1 Yêu cầu chức năng 1.1

ID: FR1

Tên: Tải về ứng dụng

DESC: Người sử dụng có khả năng tải về wusng dụng di động qua kho ứng dụng hoặc dịch vụ tương tự trên điện thoại di động. Ứng dụng là miễn phí để tải về.

RAT: để người sử dụng tải ứng dụng di động về.

DEP: không

3.2.1.2 Yêu cầu chức năng 1.2

ID:FR2

Tên: Ứng dụng di động - Đăng ký người sử dụng

DESC: Giả thiết là người sử dụng đã tải về và cài đặt xong ứng dụng, khi đó người sử dụng cần phải đăng ký. Người sử dụng cần cung cấp user's name, password và địa chỉ email. Và người sử dụng sẽ lựa chọn một số điện thoại mà dùng thường xuyên.

RAT: để người sử dụng đăng nhập vào ứng dụng

DEP: FR1.

3.2.1.3 Yêu cầu chức năng 1.3

ID: FR3

Tên: Ứng dụng di động - Đăng nhập

DESC: Giả sử người sử dụng đã đăng ký, khi đó người sử dụng cần phải đăng nhập vào ứng dụng di động. Thông tin đăng nhập được lưu trong điện thoại và tương lai người sử dụng sẽ được đăng nhập tự động.

RAT: để người sử dụng có thể đăng nhập vào ứng dụng di động.

DEP: FR1, FR2.

3.2.1.4 Yêu cầu chức năng 1.4

ID: FR4

Tên: Ứng dụng di động - Tìm kiếm nhà hàng

DESC: Giả sử người sử dụng đã đăng nhập vào ứng dụng di động, khi đó trang đầu được đưa ra là trang tìm kiếm. Người sử dụng cần tìm một nhà hàng, tùy theo một số lựa chọn tìm kiếm. Các lựa chọn tìm kiếm là Giá, Khoảng cách, Kiểu nhà hàng và Món ăn chuyên biệt. Cũng cần có lựa chọn tìm kiếm là văn bản nhập tự do. Người sử dụng có khả năng lựa chọn nhiều tiêu chí tìm kiếm trong một lần tìm.

RAT: Để người sử dụng tìm kiếm nhà hàng

DEP: FR3.

3.2.1.5 Yêu cầu chức năng 1.5

ID: FR5

Tên: Ứng dụng di động - Hiển thị kết quả tìm kiếm dạng xem bản đồ

DESC:

- Kết quả tìm kiếm có thể xem trên bản đồ. Trên bản đồ, các nhà hàng liên quan và gần nhất theo vị trí của người sử dụng được chỉ ra.
- Một ghim chuyên biệt sẽ hiển thị vị trí nhà hàng. Trên mỗi ghim cần có đường link thông tin.
- Cần có tối đa 100 kết quả được hiển thị. Xem dạng bản đồ có độ phóng mặc định.
- Xem dạng bản đồ có chứa nút bấm, khi được lựa chọn, cần hiển thị các lựa chọn lọc khác nhau trên menu lọc.

RAT: Cách kết quả được hiển thị trên bản đồ.

DEP: FR4.

3.2.1.6 Yêu cầu chức năng 1.6

ID: FR6

Tên: Ứng dụng di động - Hiển thị kết quả dạng danh sách

DESC:

- Kết quả tìm kiếm có thể được xem dạng danh sách. Mỗi phần tử trong danh sách thể hiện một nhà hàng cụ thể. Mỗi phần tử cần chứa tên, số điện thoại, kiểu món ăn, khoảng cách đến vị trí người dùng, giá trung bình, mô tả ngắn 2 dòng, đường link đến trang Web của nhà hàng và đường link thông tin.
- Có tối đa 100 kết quả được hiển thị. Nếu kết quả chứa nhiều hơn 100 nhà hàng, người sử dụng có thể cuộn dọc theo chúng.
- Khi tìm kiếm theo giá các nhà hàng cần được sắp xếp theo thứ tự sau:
 1. Giá trung bình
 2. Khoảng cách
 3. Kiểu nhà hàng
 4. Món ăn chuyên biệt
- Khi tìm kiếm theo lựa chọn tìm kiếm, khác với giá, các nhà hàng cần được sắp xếp theo thứ tự sau:
 1. Khoảng cách
 2. Giá trung bình
 3. Kiểu nhà hàng
 4. Món ăn chuyên biệt
- Xem dạng danh sách cần chứa tiêu đề của các lựa chọn tìm kiếm khác nhau.
- Xem dạng danh sách cần chứa nút bấm, khi lựa chọn, cần hiển thị các lựa chọn khác nhau trong menu lọc.

RAT: Cách kết quả được hiển thị dạng danh sách.

DEP: FR4.

3.2.1.7 Yêu cầu chức năng 1.7

ID: FR7

Tên: Ứng dụng di động – Định hướng đến nhà hàng

DESC: Mỗi người sử dụng cần chọn ghim trên bản đồ hoặc một phần tử trong danh sách. Khi lựa chọn được thực hiện, vị trí của nhà hàng cần được gửi đến chương trình định hướng GPS của điện thoại di động. Người sử dụng sau đó cần được hỗ trợ chỉ đường đến đích. Khi đến đích, người sử dụng có khả năng trở lại trang web tìm kiếm trên ứng dụng di động.

RAT: Chỉ đường cho người sử dụng đến nhà hàng đã chọn.

DEP: FR5, FR6.

3.2.1.8 Yêu cầu chức năng 1.8

ID: FR12

Tên: Ứng dụng di động – Tìm theo giá

DESC: Người sử dụng có khả năng nhập khung giá thấp nhất và cao nhất. Kết quả sẽ được hiển thị danh sách xem theo mặc định.

RAT: Để người sử dụng tìm kiếm theo giá

DEP: FR4.

3.2.1.9 Yêu cầu chức năng 1.9

ID: FR9

Tên: Tìm kiếm theo khoảng cách.

DESC: Người sử dụng cần có khả năng nhập khoảng cách tối thiểu và tối đa đến vị trí của họ. Mặc định khoảng cách tối thiểu được đặt là 0km và tối đa là 10km. Kết quả được hiển thị trên bản đồ theo mặc định.

DEP: FR4.

3.2.1.10 Yêu cầu chức năng 1.10

ID: FR10

Tên: Tìm kiếm theo kiểu nhà hàng

DESC: Người sử dụng có thể lựa chọn kiểu nhà hàng trong danh sách cho trước như đầu vào. Kết quả hiển thị trên bản đồ theo mặc định.

DEP: FR4

3.2.1.11 Yêu cầu chức năng 1.11

ID: FR11

Tên: Tìm kiếm theo món ăn

DESC: Người sử dụng có khả năng chọn món ăn cụ thể trong danh sách cho trước như đầu vào. Kết quả hiển thị trên bản đồ theo mặc định.

RAT: Để người sử dụng tìm kiếm theo món ăn cụ thể.

DEP: FR4.

3.2.1.12 Yêu cầu chức năng 1.12

ID: FR12

Tên: Ứng dụng di động – Tìm kiếm theo văn bản tự do

DESC: Người sử dụng có khả năng tìm kiếm bằng cách nhập tên nhà hàng, mô tả nhà hàng, địa chỉ nhà hàng, kiểu nhà hàng hoặc thực đơn nhà hàng ở trường văn bản tự do. Kết quả được hiển thị trên bản đồ theo mặc định.

RAT: Để người sử dụng tìm kiếm theo nhập văn bản tự do.

DEP: FR4.

3.2.1.13 Yêu cầu chức năng 1.13

ID: FR13

Tên:

DESC: Khi xem kết quả trên danh sách hoặc bản đồ, người sử dụng có khả năng lọc kết quả trong thực đơn lọc. Các lựa chọn lọc là:

- Tăng hay giảm khoảng cách tối đa
- Tăng hay giảm giá tối đa
- Chọn kiểu nhà hàng
- Chọn món ăn cụ thể

Khi lọc kết quả, chỉ các kết quả đang có mới chịu tác động và truy vấn tìm kiếm mới không cần được gửi.

RAT: Để cho người sử dụng lọc kết quả trên danh sách hoặc bản đồ.

DEP: FR5, FR6.

3.2.2 Lớp người sử dụng 2 – Chủ nhà hàng

3.2.2.1 Yêu cầu chức năng 2.1

ID: FR 14

Tên: Ứng dụng Web - Tạo tài khoản

DESC: Để tạo một tài khoản, chủ nhà hàng cần đăng ký với web portal.

- Kịch bản: Thông tin yêu cầu để đăng ký

Cho trước chủ nhà hàng muốn tạo tài khoản và chủ nhà hàng chưa có tài khoản. Chủ nhà hàng đăng ký trên web portal bằng việc cung cấp user's name, password, address, email address, phone number. Sau đó chủ nhà hàng cần áp vào để kiểm chứng.

RAT: Để chủ nhà hàng đăng ký

DEP: không

3.2.2.2 Yêu cầu chức năng 2.2

ID: FR 15

Tên: Chủ nhà hàng đăng nhập

DESC: Để sử dụng hệ thống, chủ nhà hàng cần đăng nhập vào web portal

Cho trước chủ nhà hàng muốn đăng nhập với tên của mình, anh ta cần nhập mật khẩu.

RAT: Để chủ nhà hàng đăng nhập vào web portal

DEP: FR 14.

3.2.2.3 Yêu cầu chức năng 2.3

ID: FR 16

Tên: Chủ nhà hàng nhập thông tin nhà hàng

DESC: Để sử dụng hệ thống, chủ nhà hàng cần đăng nhập vào web portal

- Kịch bản: Điền vào các ô bắt buộc

Chủ nhà hàng nhập giá trung bình, địa chỉ, địa chỉ email, số điện thoại

- Kịch bản: Điền vào các ô tự chọn

Chủ nhà hàng cung cấp: mô tả nhà hàng, thực đơn, kiểu nhà hàng, ảnh nhà hàng, số điện thoại di động

- Kịch bản: Điền vào các ô thực đơn.

Chủ nhà hàng nhập tên món ăn, mô tả món ăn, giá món ăn và tiếp tục các món ăn khác.

- Kịch bản: Sửa hoặc xóa các thông tin

RAT: Để chủ nhà hàng nhập thông tin nhà hàng trên web portal

DEP: FR 15.

Và các yêu cầu chức năng với người quản trị như: Kiểm chứng thông tin chủ nhà hàng, cập nhật tài khoản chủ nhà hàng, cập nhật kiểu nhà hàng, cập nhật món ăn, kiểm chứng thông tin nhà hàng, quản trị thông tin nhà hàng, cập nhật người dùng di động.

Bài 2: Đặc tả thiết kế phần mềm

Đặc tả thiết kế Website bán hàng trực tuyến

Tham khảo tài liệu:

Software Design Document for Online Shopping Mall TEXAS A&M UNIVERSITY

Prepared by:

Ross Dusenbury,
Bhawana Goel,
Dongkun Kim,
Tarun Kumar Nandi Suresh Babu,
Yuanling Li,
Hongtao Quin,
Vijaya Senthil Veeri Chetty

Mục lục

1. Mở đầu
 - 1.1 Mục đích
 - 1.2 Phạm vi dự án
 - 1.3 Công nghệ sử dụng
2. Mô tả chung
 - 2.1 Tổng quan về sản phẩm
 - 2.2 Các chức năng
 - 2.2.1 Cho người sử dụng
 - 2.2.2 Cho người bán
 - 2.2.3 Cho người quản trị
 - 2.3 Các lớp người dùng và các đặc tính
 - 2.4 Các ràng buộc thiết kế và cài đặt
3. Các ca sử dụng và luồng hoạt động (Use cases và Activity flow)
4. Phân tích hệ thống
 - 4.1 Lược đồ ER
 - 4.2 Lược đồ luồng dữ liệu
5. Các lớp

- 5.1 Cấu trúc dữ liệu bên trong
- 6. Kiến trúc hệ thống
- 7. Giao diện người sử dụng
- 8. Kiểm thử
 - 8.1 Log in
 - 8.2 Log in và Bắt đầu mua bán
 - 8.3 Xem các hỗ trợ trực tuyến
 - 8.4 Trả tiền
 - 8.5 Chuyển hàng
 - 8.6 Log out
- 9. Kết luận
- 10. Tài liệu tham khảo

1. Mở đầu

Tài liệu này là báo cáo thiết kế về ứng dụng Mua bán trực tuyến trên nền Web. Nó chủ yếu về “làm như thế nào” và cũng hỗ trợ cung cấp cái nhìn bên trong cho thiết kế và cài đặt toàn bộ hệ thống Mua bán trực tuyến. Phần mềm có ba thành phần chính sau:

- a. Cài đặt các kiểu người sử dụng khác nhau: Quản trị, Khách hàng và Nhà cung cấp
- b. Cài đặt giỏ hàng cho các thủ tục thanh toán và trả tiền
- c. Quản lý sản phẩm và các đơn đặt hàng

Tài liệu thiết kế này bao gồm chính là các lược đồ Máy trạng thái/ Luồng hoạt động, Thiết kế Class, Cấu trúc dữ liệu bên trong, Thiết kế kiến trúc, Giao diện người dùng và Kiểm thử. Các mục đích chính của tài liệu thiết kế được liệt kê dưới đây.

- a. Hiểu chính xác các yêu cầu và ràng buộc liên quan đến ngôn ngữ lập trình và giao diện người dùng
- b. Phân rã hệ thống thành các đơn vị hoặc module quản trị được
- c. Trừu tượng cài đặt hệ thống với sự trợ giúp của các lớp
- d. Cung cấp phác thảo Giao diện người dùng của hệ thống Mua bán trực tuyến

Báo cáo này là kết quả của giai đoạn thiết kế. Các lược đồ lớp và hoạt động cung cấp thể hiện qua sơ đồ các lớp khác nhau được sử dụng và luồng thiết kế tương ứng. Các lược đồ này đã được thiết lập sử dụng kiến trúc phần mềm quan hệ IBM. Lược đồ phối hợp hoạt động cung cấp mô hình để trực quan luồng trong hệ thống như một hoạt động tổng thể. Các lược đồ lớp cũng chứa các cấu trúc dữ liệu mà có thể được sử dụng và các kiểu dữ liệu của chúng.

Trang Web mua bán được cài đặt sử dụng PHP như một ngôn ngữ lập trình, MySQL được sử dụng để lưu giữ thông tin người dùng, sản phẩm và đơn đặt hàng.

1.1 Mục đích

Mục đích của dự án này là tạo ra một Website ứng dụng cho người sử dụng có thể cả mua và bán hàng hóa. Người sử dụng website này sẽ có thể tạo tài khoản mà sẽ làm cho họ dễ dàng sử dụng giao diện để cả trả tiền cho các hàng hóa được liệt kê bởi người sử dụng khác và bán các

đồ vật cho người sử dụng khác. Các yêu cầu chính khẳng định rằng người sử dụng có khả năng đăng ký và quản lý hồ sơ của họ một cách an toàn. Họ sẽ có khả năng điều hướng dọc theo các đồ vật sẵn có qua chủng loại hoặc sử dụng hàm tìm kiếm thực thi tốt. Cuối cùng, cơ chế trả tiền sẽ được xây dựng để cho phép sử dụng thẻ tín dụng và séc để trả tiền các đồ vật trong đơn hàng theo một cách khái quát của phần khó nhất trong hệ thống bán hàng trực tuyến. Website sẽ được bảo trì và quản lý bởi những người quản trị, Họ sẽ có chủ quyền để duyệt hoặc từ chối các đơn mua mới và bảo trì các danh sách khác nhau của các chủng loại hàng hóa. Các quản trị viên cũng có khả năng kiểm soát các khách hàng và các hoạt động của khách vắng lại.

1.2 Phạm vi dự án

Đối với dự án này chúng ta muốn xác định cái gì sẽ làm được như sản phẩm cuối cùng đáp ứng được các mong đợi. Các phần sau sẽ được hoàn thành:

- Đăng ký và các tiện ích quản lý hồ sơ an toàn cho khách hàng
- Trình duyệt qua Cửa hàng trực tuyến để xem các đồ vật có trong mỗi chủng loại hàng hóa như đồ dùng nhà bếp, đồ dùng nhà tắm, đồ ăn, ...
- Các cơ chế tìm kiếm thích hợp để dễ dàng và nhanh chóng truy cập các sản phẩm hoặc dịch vụ cụ thể
- Cơ chế và cổng trả tiền cho mọi loại thẻ tín dụng và séc thông dụng và các lựa chọn trả tiền liên quan khác.

Bổ sung vào các yêu cầu chức năng cơ bản đã nêu trên cho dự án, chúng ta lập kế hoạch cho các yêu cầu thêm như sau. Chúng ta sẽ tạo giỏ hàng sao cho khách hàng có thể lưu giữ các đồ vật của họ và thanh toán sau khi chọn xong toàn bộ giỏ hàng. Một cơ chế sắp xếp cho sản phẩm trong từng chủng loại như: được mua nhiều nhất, giá từ thấp đến cao, và liệt kê hàng mới. Chúng ta cũng lập kế hoạch cho cơ chế phản hồi cho sản phẩm từ các khách hàng. Phản hồi cũng cần được cho đối với nhà cung cấp cụ thể và toàn bộ cửa hàng.

Để duy trì hoạt động của hệ thống và cũng như tính an toàn của khách hàng và nhà cung cấp, chúng ta sẽ đảm bảo các điều sau trong dự án:

- Chất lượng toàn bộ hệ thống cần đảm bảo tốt và nó cần hoạt động trơn tru cho cả nhà cung cấp và khách hàng.
- Hệ thống cần dễ dàng bảo trì. Chúng ta sẽ sử dụng các kỹ thuật lập trình hướng đối tượng.
- Chi tiết của khách hàng và nhà cung cấp sẽ được bảo mật chống kẻ xâm nhập bên ngoài.
- Các cổng trả tiền an toàn sẽ được sử dụng và mọi kiểm tra an ninh sẽ được thực hiện đảm bảo mua bán an toàn.
- Hệ thống sẽ sẵn sàng mọi thời điểm. Phần cứng và phần mềm cần phải hoạt động tốt và sẽ được kiểm tra tăng cường.

1.3 Công nghệ sử dụng

PHP – ngôn ngữ lập trình cho website

Eclipse – môi trường phát triển tích hợp được sử dụng về viết và sửa code
MySQL – Hệ quản trị cơ sở dữ liệu

2. Mô tả chung

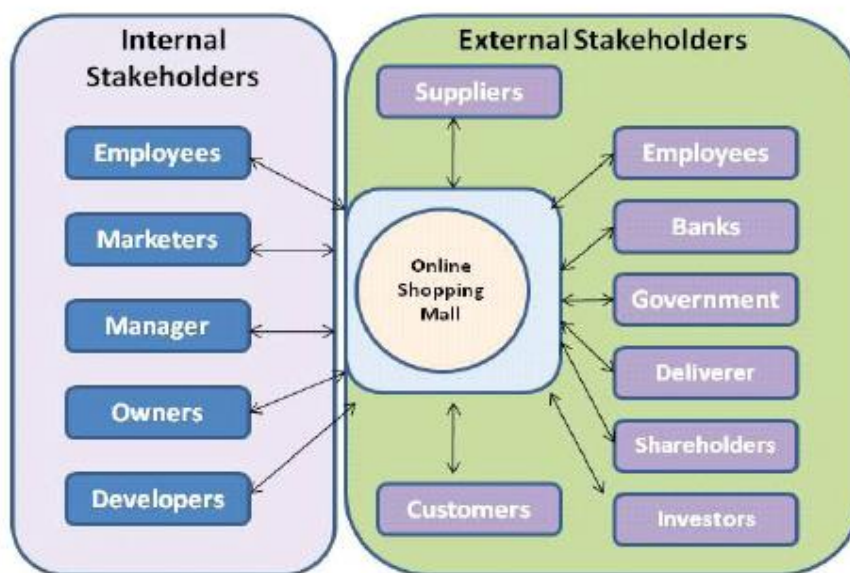
2.1 Tổng quan về sản phẩm

Cửa hàng trực tuyến sẽ cung cấp truy cập dễ dàng cho khách hàng đến các hàng hóa từ các nhà cung cấp khác nhau mà không cần đi vào gian hàng cụ thể nào. Nó cũng sẽ giúp các nhà cung cấp tìm kiếm nhiều hơn khách hàng qua Internet. Nó sẽ là một tình huống win-win đối với tất cả. Phần mềm này sẽ được thiết kế cho dễ dàng sử dụng với giao diện người dùng tốt và tin cậy. Nó sẽ đảm bảo sẵn sàng cho mọi khách hàng và nhà cung cấp ở mọi thời điểm. Hỗ trợ đó sẽ được bổ sung cho mọi hệ điều hành (Windows / Unix / Mac).

Cửa hàng trực tuyến có thể trở thành một kết nối quan trọng giữa sản phẩm và khách hàng. Trong dự án này, chúng ta sẽ phát triển một hệ thống của hàng trực tuyến cho mọi cổ đông của chúng ta. Ở đây sẽ có hai tập cổ đông liên quan: tập bên ngoài và tập bên trong.

Tập bên ngoài bao gồm các nhà cung cấp, các ngân hàng, chính quyền, các nhà phân phối, cổ đông, các nhà đầu tư và khách hàng. Yêu cầu chính là trường hợp ở đó họ có thể tìm kiếm, vận chuyển và trả tiền các đồ vật cần thiết.

Tập bên trong gồm các nhân viên, quảng cáo viên, quản trị viên, người chủ sở hữu và các người phát triển. Yêu cầu chính của họ có thể là lợi nhuận, cả ngắn hạn và dài hạn. Với các lý do đó, mỗi nhóm liên quan muốn website tập trung các đồ vật có lợi nhuận cao. Các quản trị viên sẽ đòi hỏi ứng dụng theo dõi các khách vắng lai. Các người phát triển cũng là nhóm người liên quan: họ muốn sử dụng công nghệ phát triển mới, tân tiến. Đội của chúng ta cố gắng cải tiến mọi lĩnh vực kinh doanh liên quan sao cho các thành viên thu lợi từ các dịch vụ và nhận được giá trị tốt nhất cho đầu tư của họ.



2.2 Các chức năng của sản phẩm

2.2.1 Đối với người sử dụng

Các đặc trưng tài khoản cơ bản được yêu cầu cho cả khách hàng và người bán. Nó bao gồm:

- Đăng ký
- Đăng nhập
- Quên/Thay đổi mật khẩu
- Mục “Chi tiết tài khoản” chứa chi tiết thông tin liên hệ
- Duyệt và Tìm kiếm
- Người sử dụng cần có khả năng duyệt dọc theo toàn bộ danh sách đồ vật bằng việc điều chỉnh tiêu chí lựa chọn như: chủng loại hàng hóa, chọn khung giá, chọn dựa trên đánh giá hàng hóa bán
- Ngoài việc cung cấp đặc trưng tìm kiếm như trên, người sử dụng cần có khả năng tìm kiếm sản phẩm cụ thể
- Nó được áp dụng cho mọi người (Khách vãng lai, khách hàng, người bán và người quản trị)
- Mua bán:
 - Thêm / bỏ bớt đồ vật vào giỏ hàng của họ
 - Khách hàng có lựa chọn mua các đồ vật mong muốn được bán bởi một hoặc nhiều người bán
 - Website cần hỗ trợ mọi thể tín dụng thông dụng và đảm bảo là giao dịch được thực hiện an toàn
- Lịch sử:
 - Lịch sử mua bán: Danh sách đồ vật mua bán trước đó
 - Lịch sử xem đồ vật: Danh sách đồ vật mới xem gần nhất
 - Lịch sử tìm kiếm: tiêu chí tìm kiếm gần nhất được sử dụng bởi khách hàng

2.2.2 Đối với người bán hàng

- Thiết lập tài khoản
- Các yêu cầu chuyên biệt cho mục Tài khoản trong yêu cầu của người sử dụng
- Yêu cầu khởi tạo gian hàng
- Người bán cần có ủy quyền của Người quản trị trước khi bán sản phẩm / khởi tạo gian hàng trong Cửa hàng
- Lịch sử giao dịch
- Trạng thái của mọi giao dịch đối với một người bán

2.2.3 Đối với người quản trị

- Thiết lập Tài khoản
- Xác lập chủ quyền cho các yêu cầu khởi tạo gian hàng
- Cập nhật thông tin sản phẩm
- Giám sát các giao dịch và theo dõi trạng thái của chúng
- Cập nhật thông tin người bán/ khách hàng /giao dịch (trong trường hợp có lỗi/thắc mắc của hệ thống)

2.3 Các lớp người sử dụng và các đặc trưng

Hệ thống được cài đặt cho những người sử dụng sau:

- Cho khách hàng / khách vắng lai

Khách hàng có khả năng mua được các đồ vật thông qua việc mua ở Cửa hàng. Cô ta có khả năng nhìn thấy lịch sử mua của mình. Một phiên an toàn sẽ được duy trì và tên người sử dụng và mật khẩu sẽ được chỉ định rõ cho mỗi khách hàng. Các khách hàng không có khả năng tạo nên sự thay đổi cho hệ thống. Cô ta chỉ có khả năng duyệt qua các đồ vật sử dụng hàm tìm kiếm và đặt mua chúng. Cô ta có thể sau đó trả tiền trực tuyến thông qua phương pháp trả tiền an toàn và cung cấp địa chỉ giao dịch và địa chỉ chuyển hàng. Khách hàng sẽ cũng có thể đánh giá các đồ vật khác nhau và xếp hạng chúng.

- Cho người quản trị

Một người quản trị có nhiều quyền hơn người sử dụng bình thường. Các chức năng chính của người quản trị là quản lý khách hàng và các nhà cung cấp. Người quản trị có thể xem xét, sửa và xóa thông tin cá nhân và mật khẩu của các thành viên nếu cần thiết. Họ cũng có khả năng tìm kiếm thông tin về các thành viên và các nhà cung cấp. Các người quản trị cũng có khả năng xem xét danh sách các hóa đơn của các đơn đặt, các đơn chuyển hàng và các đơn bị hủy.

- Cho các nhà cung cấp

Mỗi người sử dụng có thể đăng ký yêu cầu tạo gian hàng thông qua ứng dụng. Sau khi yêu cầu được chấp nhận bởi người quản trị Cửa hàng, người yêu cầu sẽ được thông báo. Chủ gian hàng sẽ có trách nhiệm thiết lập các sản phẩm và quản trị gian hàng của họ. Chủ gian hàng có thể thêm, xóa, cập nhật các đồ vật cho gian hàng. Cô ta sẽ cần gán đồ vật vào danh sách chủng loại của người quản trị sao cho việc quản lý sản phẩm trong Cửa hàng được dễ dàng. Chủ gian hàng cũng có thể quyết định đóng gian hàng và xóa nó khỏi Cửa hàng.

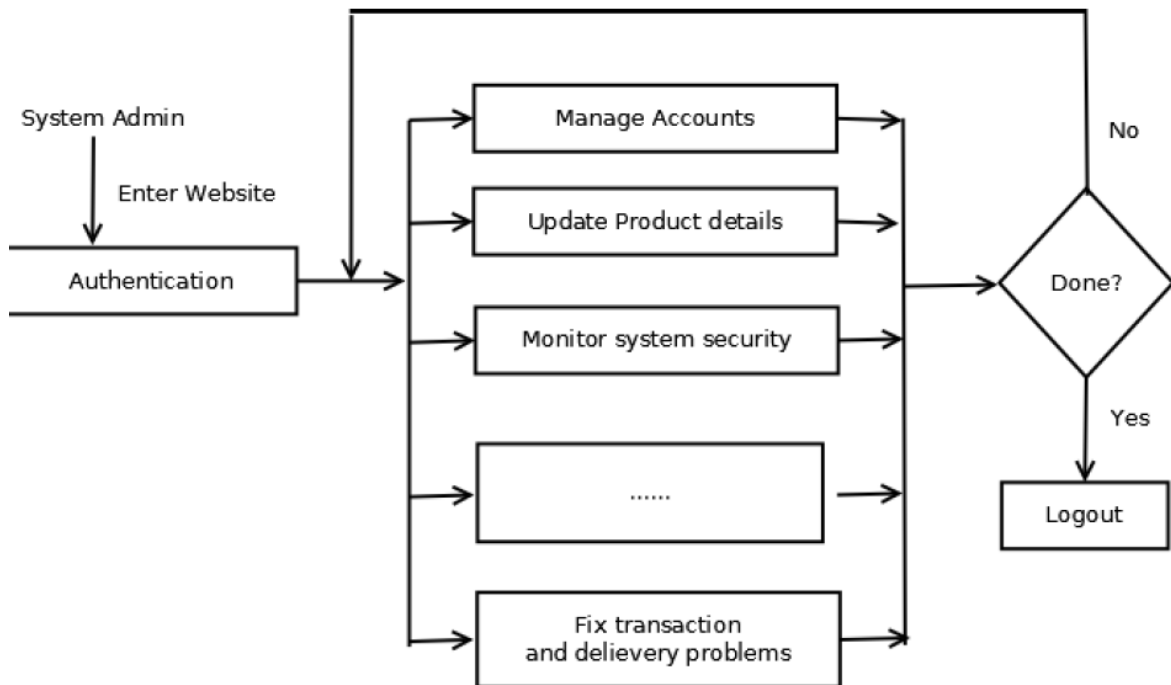
2.4 Các ràng buộc thiết kế và cài đặt

Để tạo phần mềm tốt, các ràng buộc thiết kế và các tiếp giáp khác cần được quan tâm. Trong mục này chúng ta sẽ liệt kê tất cả xem xét mà cần được tạo khi hoàn thành dự án.

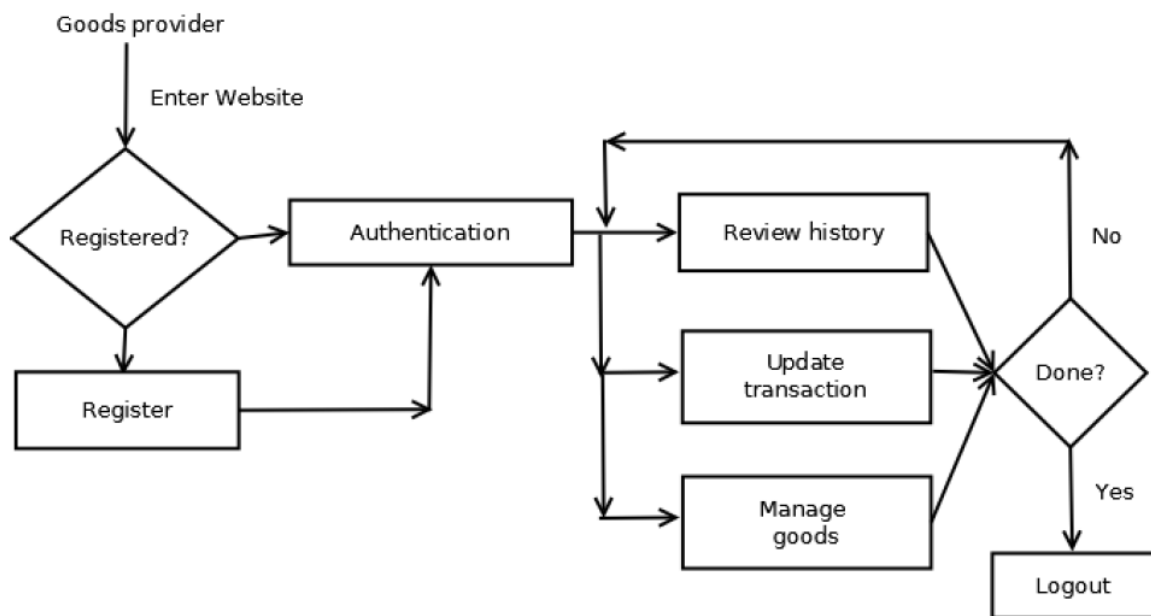
- Cần được code đủ hiệu quả để chạy tốt trên phần cứng server được cung cấp
- Code phía client và / hoặc trang web cần có khả năng chạy hiệu quả trên phần cứng đầu cuối thấp
- Cơ sở dữ liệu sẽ được tạo và bảo trì theo cách mà làm cho nó có kích thước chấp nhận và quản trị được.

3. Các ca sử dụng và luồng hoạt động

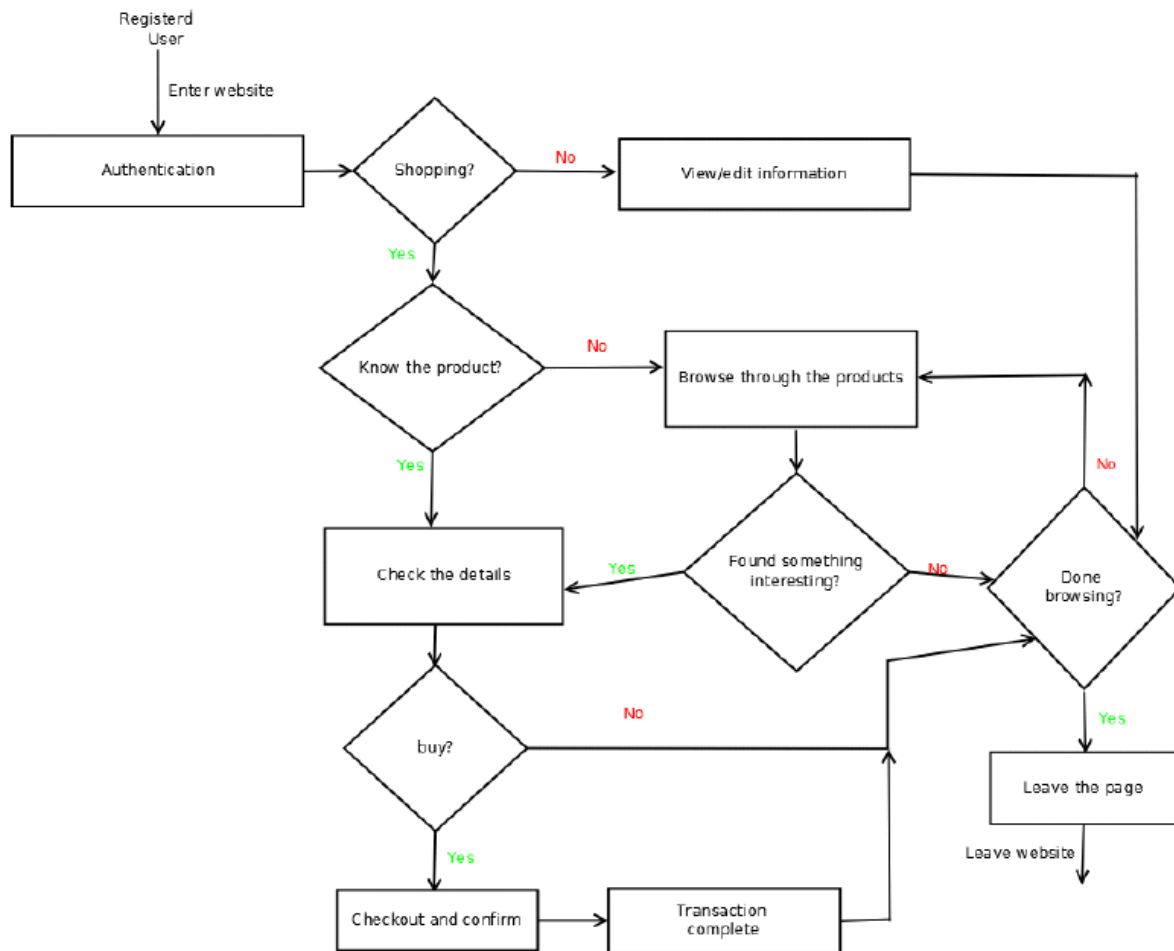
Luồng hoạt động gắn với quản trị hệ thống



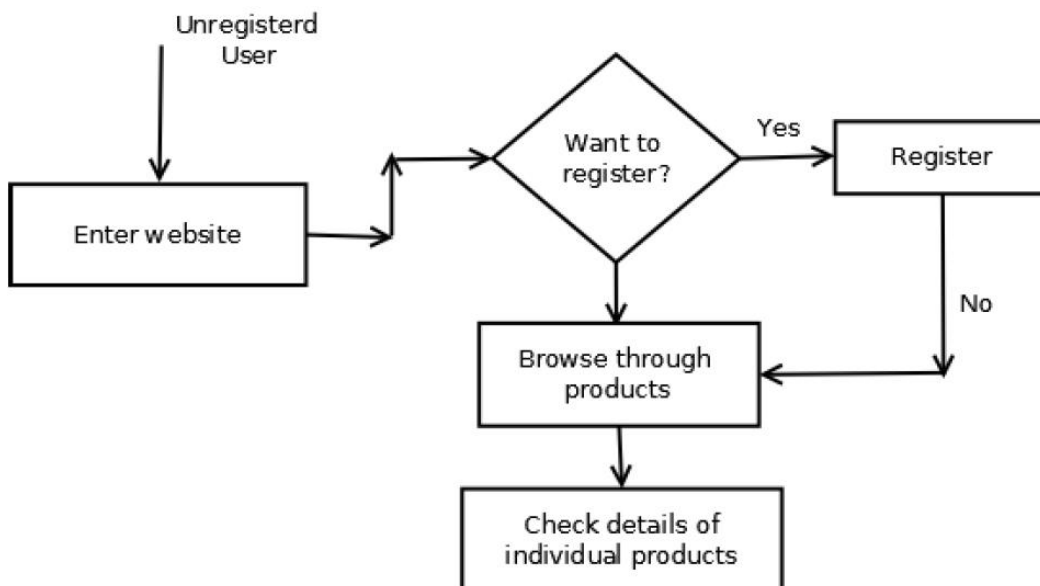
Luồng hoạt động gắn với chủ gian hàng



Luồng hoạt động gắn với người sử dụng đã đăng ký

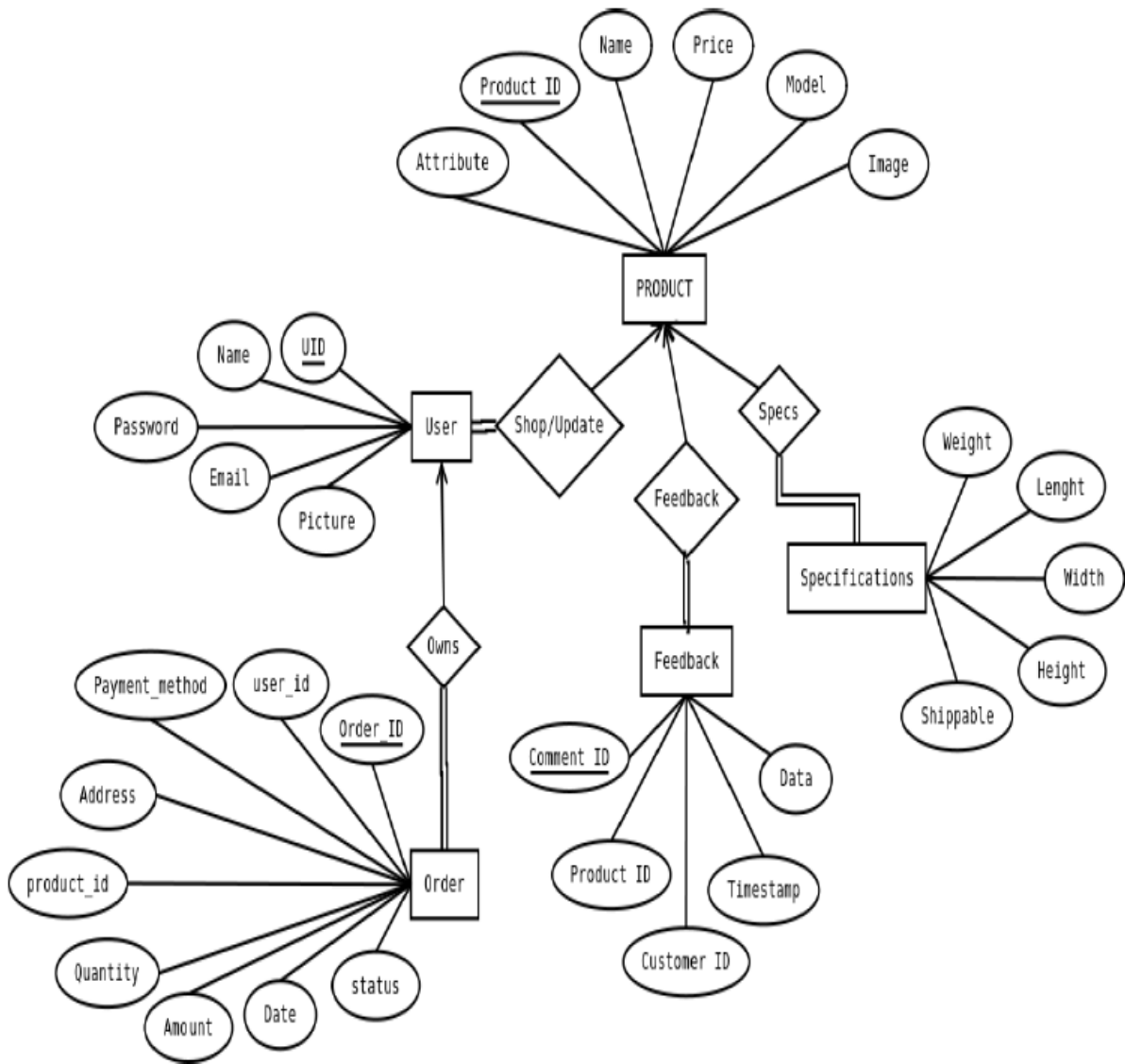


Luồng hoạt động gắn với người sử dụng chưa đăng ký



4. Phân tích hệ thống

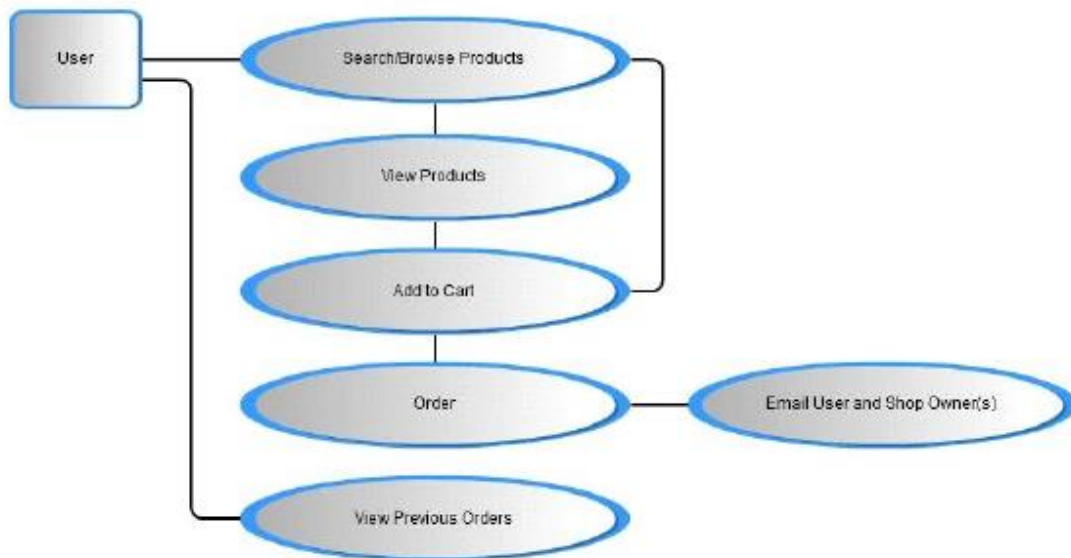
4.1 Lược đồ quan hệ thực thể



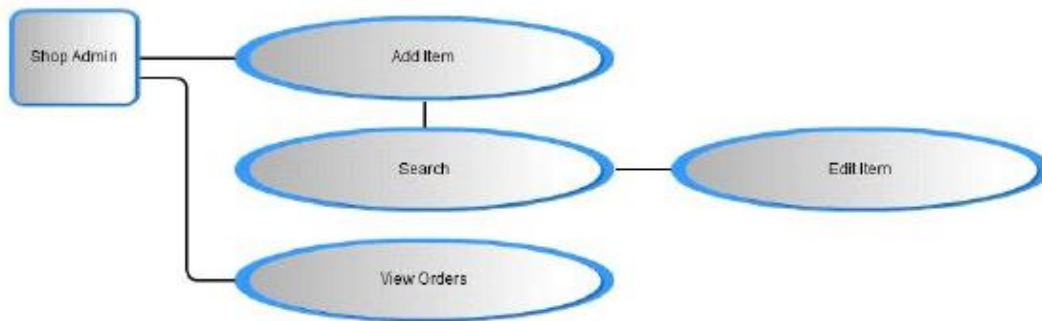
4.2 Lược đồ luồng dữ liệu

Các lược đồ luồng dữ liệu này biểu diễn thông tin được di chuyển như thế nào từ hành động này sang hành động khác cho một nhóm người sử dụng. Chúng chỉ ra thông tin đi như thế nào cho mỗi người sử dụng và các đường đi mà họ thực hiện như họ hoàn thiện các hành động khác nhau chuyên biệt cho luồng công việc của họ.

Người sử dụng:



Chủ gian hàng:

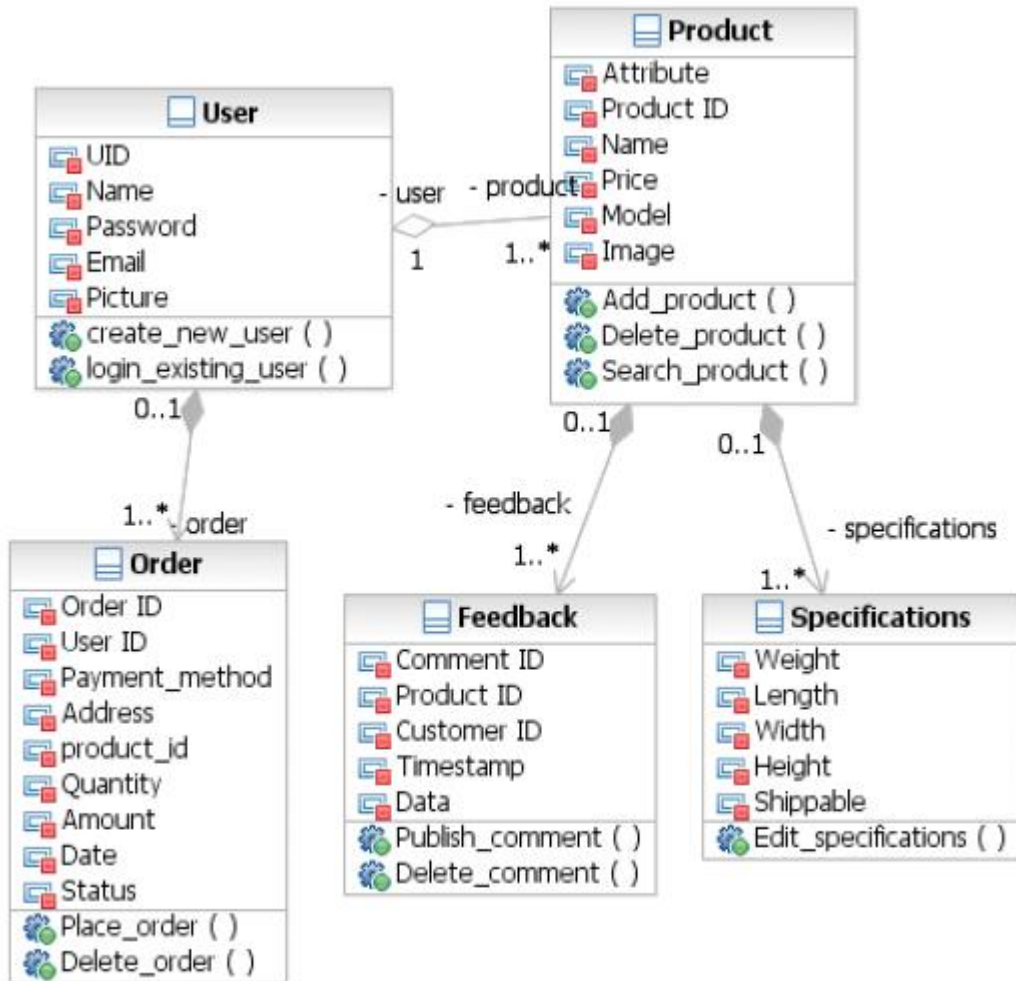


Người quản trị:



5. Các lớp

Các lớp sau được sử dụng để thiết kế Cửa hàng mua bán trực tuyến. Lược đồ lớp được sử dụng để thể hiện quan hệ giữa các lớp trong UML.



Lược đồ lớp

Lớp User

Lớp User được sử dụng để lưu giữ chi tiết của mọi người sử dụng. Nó được sử dụng để tạo người sử dụng mới hoặc hỗ trợ người sử dụng đã có đăng nhập.

Ở đây có hai loại người sử dụng: admin và user bình thường. Mỗi người sử dụng có thể đặt đơn mua hàng hoặc tìm kiếm sản phẩm. Admin có thể bổ sung hoặc xóa các user bình thường và các sản phẩm.

Lớp order

Lớp order chứa chi tiết của đơn đặt hàng. Lớp này được sử dụng để đặt hoặc xóa đơn hàng. Lớp đơn hàng được thông báo khi một đơn được đặt hoặc xóa bởi một người sử dụng

Lớp Product

Lớp Product chứa chi tiết các sản phẩm trong Cửa hàng mua bán trực tuyến. Lớp này sẽ được thông báo khi người sử dụng tìm một sản phẩm hoặc admin bổ sung hoặc xóa một sản phẩm.

Lớp Feedback

Lớp Feedback chứa chi tiết các phản hồi được đăng về các sản phẩm khác nhau. Lớp này có trách nhiệm đăng các phản hồi về các sản phẩm và xóa chúng bởi người sử dụng.

Lớp Specifications

Lớp đặc tả chứa các đặc tả khác nhau của sản phẩm. Lớp này được sử dụng để chỉnh sửa đặc tả của các sản phẩm. Để đơn giản có thể gộp vào Product class.

Lưu ý

Ở đây chúng ta có thể thêm một số lớp như: Category Class để phân loại sản phẩm; Order's Details Class để lưu giữ các mục trong giỏ hàng.

Và tùy theo mặt hàng kinh doanh có thể bổ sung thêm các thông tin một tả các lựa chọn khác nhau, thông tin bảo hành, thời hạn sử dụng và một số tính chất khác, vì mỗi loại hàng có những đặc thù riêng của nó.

5.1 Cấu trúc dữ liệu bên trong

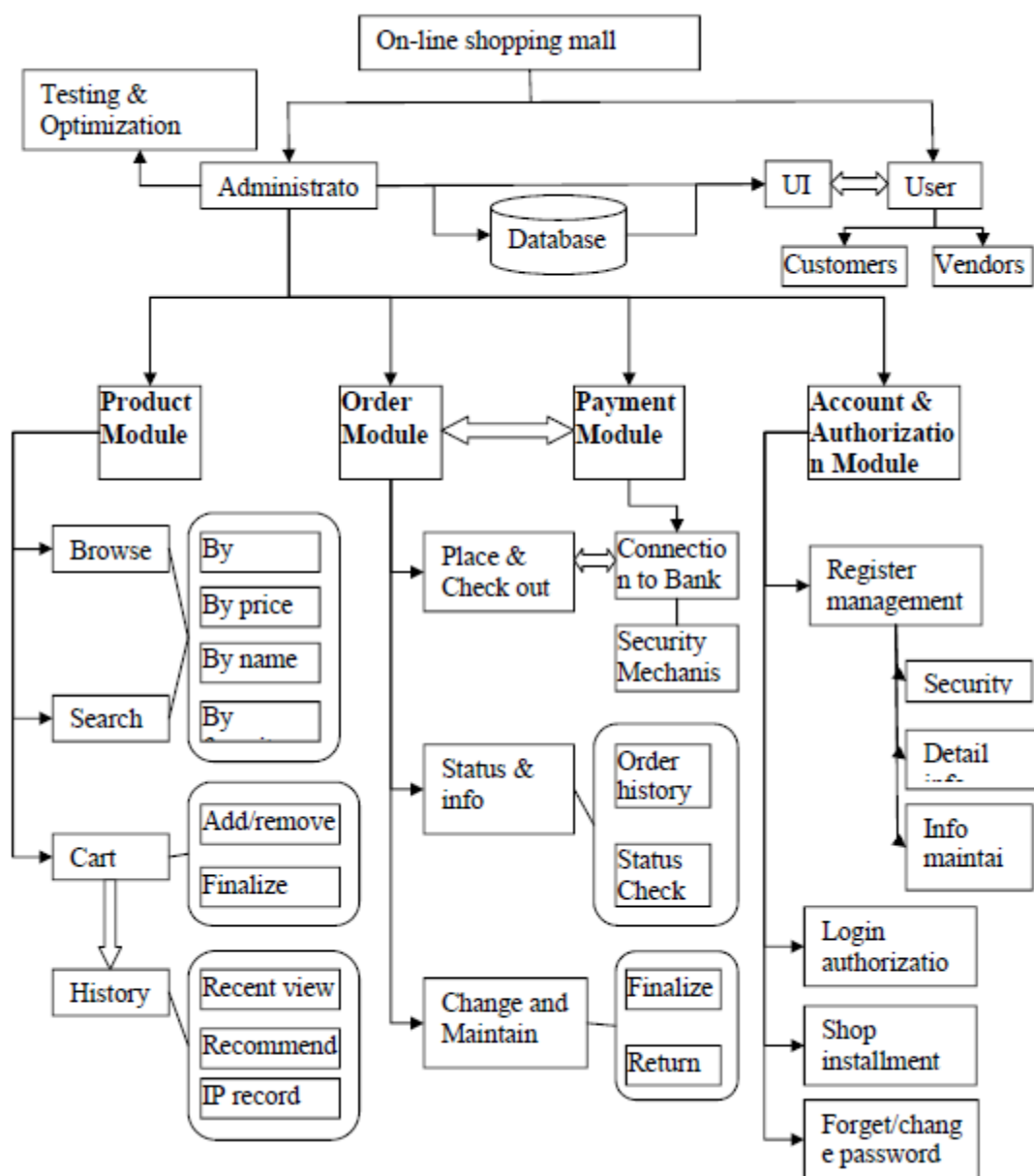
Các cấu trúc dữ liệu bên trong là các biến thành viên của mỗi lớp. Sau đây là các cấu trúc dữ liệu bên trong của mỗi lớp.

- **User Class**
 - **UID:** đây là định danh duy nhất của mỗi người sử dụng
 - **Name:** tên của người sử dụng
 - **Password:** mật khẩu của người sử dụng
 - **Email:** địa chỉ email của người sử dụng
 - **Phone:** số điện thoại của người sử dụng
 - **Picture:** ảnh upload lên của người sử dụng
- **Order class**
 - **Order ID:** mã đơn hàng gắn đơn cụ thể
 - **User ID:** ID của người sử dụng đặt đơn hàng
 - **Payment method:** hình thức trả tiền
 - **Address:** địa chỉ chuyển hàng đến
 - **Amount:** giá tiền của đơn hàng
 - **Date:** ngày đơn được đặt
 - **Status:** trạng thái hiện tại của đơn hàng: đơn hàng đang trong quá trình xử lý hay đang chuyển đến khách hàng hay đã được chuyển xong.

- **Order's Detail Class**
 - **Order's Detail ID:** định danh duy nhất của chi tiết đơn hàng tương ứng với một đơn và sản phẩm cụ thể
 - **Order ID:** ID của đơn hàng
 - **Product ID:** ID của sản phẩm mua
 - **Quantity:** số lượng sản phẩm đó trong chi tiết đơn hàng
 - **Detail's Amount:** giá chi tiết đơn hàng
- **Product Class**
 - **Product ID:** định danh duy nhất của sản phẩm
 - **Name:** tên của sản phẩm
 - **Attribute:** mô tả sản phẩm
 - **Price:** giá của sản phẩm
 - **Model:** số hiệu kiểu của sản phẩm
 - **Image:** ảnh của sản phẩm
 - **Category ID:** định danh loại sản phẩm
- **Category Class**
 - **Category ID:** định danh duy nhất của Chung loại
 - **Name:** tên của chung loại
 - **Attribute:** mô tả chung loại
 - **Image:** ảnh của chung loại
 - **Father_Cat_ID:** ID lớp cha của chung loại
- **Feedback class**
 - **Comment ID:** ID của phản hồi được đăng
 - **Product ID:** ID của sản phẩm mà về nó phản hồi phản ánh
 - **Customer ID:** ID của khách hàng viết phản hồi
 - **Timestamp:** thời gian khi phản hồi được đăng
 - **Data:** xâu văn bản được đăng coi là một phản hồi
- **Specifications Class**
 - **Weight:** trọng lượng của sản phẩm
 - **Length:** độ dài của sản phẩm
 - **Width:** độ rộng của sản phẩm
 - **Height:** chiều cao của sản phẩm
 - **Shippable:** có thể chuyển hàng được không

6. Thiết kế hệ thống

So sánh với các cấu trúc trước hoặc kiến trúc các chức năng hoặc các tác nhân, phần này chủ yếu tập trung vào kiến trúc thiết kế hệ thống tổng thể, mà mô tả các yêu cầu và các cấu trúc cần thiết bên trong theo quá trình thiết kế đối với những người thiết kế hệ thống.



Trong hình vẽ trên, “→” nghĩa là hàm số liên quan hoặc cài đặt quản trị, và “=>” nghĩa là module truyền thông liên quan.

Ngoại trừ cho Giao diện người sử dụng, kiểm thử và tối ưu hóa, Cửa hàng mua bán trực tuyến được chia thành bốn modules chính như sau. Mỗi một trong chúng được cài đặt như một tiện ích cơ bản cho hoặc các nhà cung cấp hoặc khách hàng hoặc cả hai.

- Product module
 - Một số module con của nó (hoặc các hàm) là dành riêng cho khách hàng. Module này yêu cầu người thiết kế cài đặt trình duyệt và hàm tìm kiếm cho những người viếng thăm trang, và nó được giả thiết là thân thiện và tin cậy (dịch vụ 7-24, ít khi

bị dừng và khôi phục). Như đã nói tới trong phần 2, trình duyệt/tìm kiếm cần thực hiện với các chủng loại khác nhau. Sau đó giỏ hàng phải là thuận tiện cho khách hàng và họ có thể chỉnh sửa các đồ vật lựa chọn trong danh sách giỏ hàng. Bản ghi cũ có thể thao tác như xem lại lịch sử duyệt/ tìm kiếm, đề xuất các sản phẩm tương tự,...

- Module đặt hàng Order
 - Nó gồm ba vai trò chính, thanh toán, mà kết nối khách hàng, nhà cung cấp, ngân hàng và người quản trị. Kiểm tra thông tin/trạng thái đơn hàng và bảo trì đơn hàng, mà cho phép người sử dụng tương tác (như hủy/trả lại, ...) trên các đơn hàng đã đặt đến một mức độ nào đó.
- Module Trả tiền Payment
 - Cung cấp các phương thức trả tiền (tức là các thẻ ngân hàng khác nhau hoặc các công cụ thương mại khác) và cung cấp cơ chế an ninh.
- Module Tài khoản và Chứng thực Account & Authorization
 - Phần này tạo và ghi lại thông tin người sử dụng trong cơ sở dữ liệu với các quyền và chứng thực khác nhau, mà cho phép khách hàng hoặc nhà cung cấp có tài khoản của riêng họ để mua hoặc bán. Kiến trúc liên quan là rất rõ ràng.

Mọi modules trên đều có khả năng kết nối với hệ quản trị cơ sở dữ liệu, thực hiện SQL cơ sở một cách bình thường.

7. Giao diện người sử dụng

The screenshot shows the 'User account' page on groupapa.com. The page layout includes a blue header with the site logo and navigation links like 'Go to Front' and 'See My Shopping cart'. The main content area is titled 'User account' and features a 'Create new account' button, a 'Login' link, and a 'Request new password' link. Below these links, there are two main sections: 'Account information' and 'Personal information'. The 'Account information' section contains a 'Username' field with a note that spaces are allowed but punctuation is not, and an 'E-mail address' field with a note that it's a valid e-mail address used for system notifications. The 'Personal information' section contains a 'User Name' field with a placeholder text 'Please enter your user name here'. At the bottom of the form, there is a 'Create new account' button.

Tạo tài khoản

Màn hình chính sau đăng nhập

The screenshot shows the 'Add product form' in the Groupapa.com admin interface. It includes sections for 'Product information' (name, category, price, weight, etc.) and 'Product details' (description, keywords, etc.).

Tạo sản phẩm

The screenshot shows the 'Shopping cart' in the Groupapa.com admin interface. It displays a table with columns for 'Remove', 'Products', 'Qty.', and 'Total'. The cart contains three items: 'hammer', 'PHONE4', and 'couch'. The subtotal is \$350.00. There are buttons for 'Continue shopping', 'Update cart', and 'Checkout'.

Chọn giỏ hàng

The screenshot shows the 'Store administration' interface in the Groupapa.com admin interface. It displays a grid of links for 'Orders', 'Customers', 'Products', 'Attributes', 'Reports', 'Conditional actions', 'Configuration', and 'Help'. Each link has a 'Show links' option.

Quản trị kho hàng

groupapa.com

Search this site: [Search]

Catalogs

- Phones (1)
- Furniture (1)
- Tools (2)

dongkum kim

- My account
- Create content
- Log out

some tags related products

comfortable CITY HOMER

houseware useful

more tags

Checkout

Cart contents

Qty	Products	Price
3x	IPHONE4 Color: Black	\$1,050.00
		Subtotal: \$1,050.00

Customer information

Delivery information

Billing information

Payment method

Select a payment method from the following options:

Order total preview:

Subtotal:	\$1,050.00
Order total:	\$1,050.00

Payment method:

- ☒ Check or money order
- ☐ Credit (card on a secure server)

Checks should be made out to:

Trả tiền và chuyển hàng

Thanks for your order, dongkum!

Want to manage your order online?
If you need to check the status of your order, please visit our home page at [Our store](#) and click on "My account" in the menu or login with the following link:
<http://groupapa.com/drupal/user>

Purchasing Information:

E-mail Address: dongkumkim54@gmail.com

Billing Address:
DONGKUN KIM
1501 TEXAS AVE
COLLEGE STATION, TX 77840

Shipping Address:
DONGKUN KIM
1501 TEXAS AVE
COLLEGE STATION, TX 77840

Billing Phone:

Shipping Phone:

Order Grand Total: \$1,050.00

Payment Method: Check

Order Summary:

Shipping Details:

Order #: 3

Order Date: 11/14/2010 - 23:50

Shipping Method: Standard delivery

Products Subtotal: \$1,050.00

Total for this Order: \$1,050.00

Products on order:
3 x IPHONE4 - \$1,050.00 (\$350.00 each)
SKU: LI_001
Color: Black

Where can I get help with reviewing my order?
To learn more about managing your orders on [Our store](#), please visit our [help page](#).

Please note: This e-mail message is an automated notification. Please do not reply to this message.

Thanks again for shopping with us.

[Our store](#)

Thông tin đơn hàng cuối cùng qua thông báo email

8. Kiểm thử

8.1 Kiểm thử các thành phần

Để không làm ngất quãng chức năng của hệ thống xây dựng, chúng ta sẽ kiểm thử từng ca sử dụng riêng lẻ và liệt kê các trường hợp kiểm tra cho từng ca sử dụng.

8.1.1 Log in

- a. Định danh và mật khẩu khởi tạo có đúng không, có được mã hóa không, và được kiểm chứng chưa.
- b. Kiểm tra SQL injection
- c. Kiểm tra chữ hoa, chữ thường, số, các ký tự đặc biệt trong ID và mật khẩu.
- d. Cả hai ID và mật khẩu không trống
- e. Kiểm tra ID bị trùng nhau xem có được không
- f. Kiểm tra mật khẩu yếu
- g. Kiểm tra thông báo qua email về tạo ID và khởi tạo mật khẩu

8.1.2 Đăng nhập và bắt đầu mua bán

- a. Kiểm chứng đúng khi đăng nhập trước khi bắt đầu mua bán
- b. Không thể thanh toán sản phẩm mà đã có trong khi xử lý
- c. Hiện danh sách tất cả các sản phẩm sẵn có
- d. Kiểm tra vị trí sản phẩm phiên bản mới ở trang đầu
- e. Kiểm tra số lượng đúng sản phẩm còn lại sau khi mua, bán
- f. Kiểm tra tính nhìn thấy của tìm kiếm trước
- g. Kiểm tra đặt hàng đã hết

8.1.3 Xem trợ giúp trực tuyến

- a. Tài liệu trợ giúp thích hợp với từng tình huống

8.1.4 Trả tiền

- a. Tên chứng nhận và thông tin thẻ đảm bảo an ninh
- b. Lượng tiền trả của khách hàng cho đơn đặt đúng
- c. Chứng nhận tháng năm hết hạn
- d. Chứng nhận về số CVS
- e. Kiểm tra giới hạn tối đa của thẻ tín dụng

8.1.5 Chuyển hàng

- a. Kiểm tra thông tin địa chỉ nhận
- b. Kiểm tra thông tin riêng và địa chỉ cập nhật
- c. Chuyển hàng có trả tiền và không trả tiền
- d. Kiểm tra về thời gian cho chuyển hàng
- e. Kiểm tra về phản hồi khi khách hàng không có nhà

8.1.6 Đăng xuất

- a. Lịch sử tìm kiếm có save đúng sau khi đăng xuất
- b. Kiểm tra thông tin thanh toán được lưu trong cơ sở dữ liệu
- c. Khách hàng tái điều hướng đến màn hình đăng nhập

d. Kiểm tra hệ thống email về thông tin trả tiền

8.2 Kiểm thử tích hợp

Chúng ta đã xem xét hệ thống cửa hàng mua bán trực tuyến như một hộp đen và hộp trắng; cũng kiểm tra mọi phần của nó. Tiếp theo chúng ta sẽ tiến hành theo thứ tự sau: tạo kế hoạch kiểm thử, thực hiện xem lại code của các module ứng dụng mà tích hợp tạo thành khối ứng dụng, thực thi các ca sử dụng của ứng dụng, thực hiện kiểm thử tải, kiểm thử hiệu năng, kiểm thử tổng thể và kiểm thử an ninh.

- Hệ thống có tương thích với các trình duyệt web khác nhau không (Google chrome, Fire fox, IE)
- Hệ thống có tương thích với các hệ điều hành khác nhau không (Windows, Linux và Mac)
- Hệ thống có tương thích với thao tác 32 bit hoặc 64 bit
- Chịu tải với số lượng khách hàng lớn
- Chống các tấn công an ninh

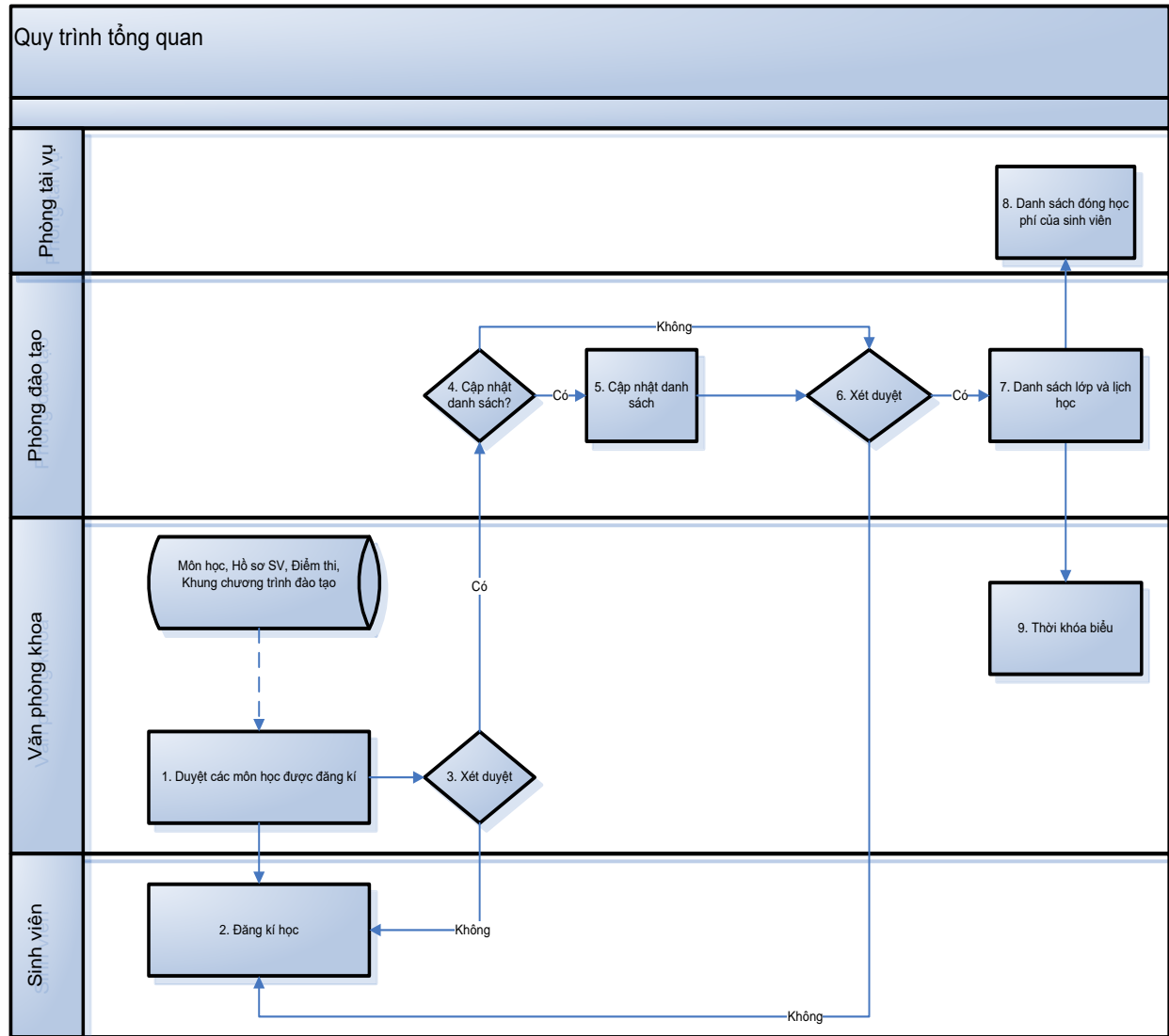
9. Kết luận

Ứng dụng có thể được sử dụng cho ứng dụng thương mại điện tử. Nó dễ sử dụng, vì nó dùng Giao diện thân thiện và tương tác mua bán trực tuyến như một hoạt động sáng tạo cho người dùng. Nó được cài đặt kiểm thử kỹ càng.

“Cửa hàng mua bán trực tuyến” được thiết kế cung cấp ứng dụng dựa trên web mà có thể tìm kiếm, xem xét và lựa chọn sản phẩm một cách dễ dàng. Cơ chế tìm kiếm cung cấp cách thuận lợi, dễ dàng để tìm kiếm sản phẩm, ở đó người sử dụng có thể tìm kiếm sản phẩm một cách tương tác và cơ chế tìm kiếm có thể lọc các sản phẩm sẵn sàng dựa trên đầu vào của người sử dụng. Người sử dụng sau đó có thể xem đặc tả đầy đủ của từng sản phẩm. Họ có thể xem các bình luận, phản hồi về sản phẩm và viết phản hồi của chính mình. Sử dụng các thành phần AJAX làm cho ứng dụng tương tác và ngăn việc tải lại khó chịu. Các đặc trưng kéo, thả của nó làm cho việc sử dụng dễ dàng hơn.

Tóm lược Đặc tả thiết kế Quản lý học tín chỉ

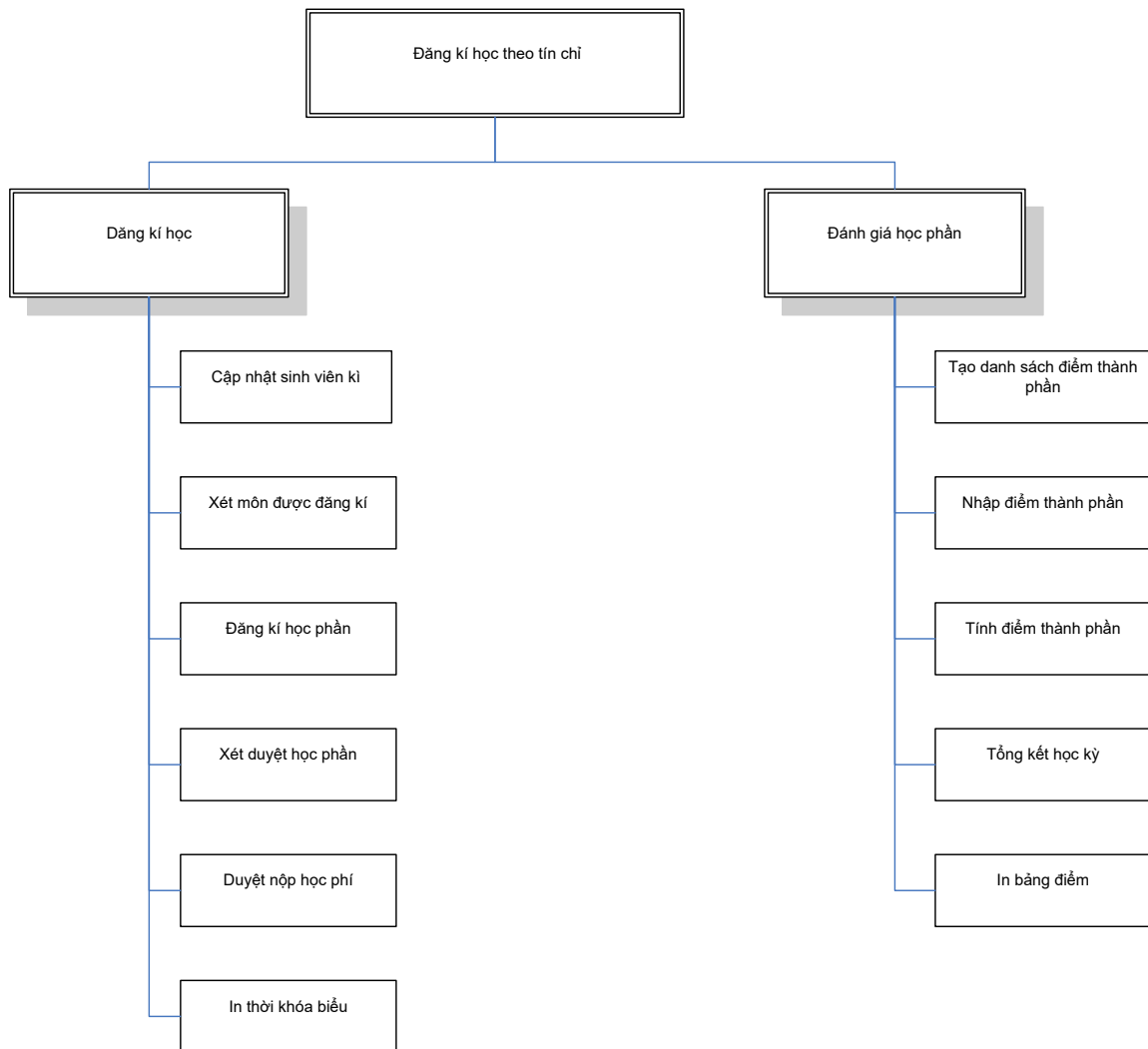
Các sơ đồ qui trình trong quá trình phân tích



Chức năng	Mô tả	Người thực hiện
1. Duyệt các môn học được đăng kí.	Trước khi sinh viên đăng kí học, khoa sẽ xét duyệt xem với mỗi sinh viên thì được đăng kí những môn học nào trong HK này. Dựa vào các tiêu chí: - Môn tiên quyết, môn trước, Ngành học...	Văn phòng khoa
2. Đăng kí học	Dựa vào những môn học đã được duyệt ở trên mà mình có thể đăng kí. Sinh viên sẽ chọn các lớp học phù hợp với mình.	Sinh viên
3. Khoa duyệt	Sau khi sinh viên đã đăng kí xong thì khoa sẽ tiến hành xét duyệt và cập nhật nếu cần thiết. Nếu được chấp nhận thì danh sách đăng kí sẽ được chuyển lên phòng đào tạo nếu không được chấp nhận thì sẽ thông báo cho sinh viên để đăng kí lại.	Văn phòng khoa.
4. Cập nhật.	Nếu cần thiết phải cập nhật lại danh sách thì tới bước 5, nếu không thì tới bước 6.	Phòng đào tạo
5. Cập nhật danh sách	Phòng đào tạo sẽ cập nhật và điều chỉnh về lịch học.	Phòng đào tạo
6. Xét duyệt	Phòng đào tạo tiến hành xét duyệt. Nếu chấp nhận thì đến bước 7. Nếu không quay trở lại bước 2.	Phòng đào tạo
7. Danh sách đăng kí	Sau khi được xét duyệt thì sẽ đưa ra được danh sách đăng kí học chính thức của sinh viên. Với các thông tin về lịch học, học phí, giáo viên dạy...	Phòng đào tạo
8. Danh sách đóng học phí	Dựa vào danh sách mà phòng đào tạo chuyển sang. Phòng tài vụ sẽ thu tiền học phí của sinh viên.	Phòng tài vụ
9. Thời khóa biểu	Dựa vào danh sách sau khi phòng đào tạo xét duyệt, sinh viên sẽ in được thời khóa biểu học của mình.	Sinh viên

Chức năng	Mô tả	Người thực hiện
1. Lựa chọn môn học	Căn cứ vào các môn học đã được khoa duyệt cho đăng kí. Sinh viên sẽ chọn môn cần đăng kí.	Sinh viên
2. Lựa chọn lớp học phần	Sinh viên cần chọn lớp học phần có lịch học phù hợp với mình.	Sinh viên
3. Kiểm tra điều kiện	Dựa vào những thông tin sinh viên đăng kí. Hệ thống sẽ kiểm tra các điều kiện như: Sĩ số, môn song hành, lịch trùng.... Nếu không thỏa mãn sẽ quay trở lại bước 1. Nếu thỏa mãn sẽ đến bước 4.	Sinh viên
4. Đăng kí	Khi đã thỏa mãn điều kiện đăng kí thì sinh viên tiến hành duyệt đăng kí môn học đó.	Sinh viên
5. Tiếp tục đăng kí	Nếu tiếp tục đăng kí thì quay lại bước 1. Ngược lại thì đến bước 6.	Sinh viên
6. Cập nhật?	Nếu sinh viên muốn cập nhật lại những môn đã đăng kí thì đến bước 7. nếu không đến bước 8.	Sinh viên
7. Sửa đổi	Sinh viên sẽ hủy môn đăng kí nếu không phù hợp và đăng kí lại môn khác.	Sinh viên
8. Danh sách đăng kí	Sau khi kết thúc đăng kí sẽ đưa ra danh sách các môn học và lớp học phần tương ứng. Và chuyển lên khoa.	Sinh viên
9. Khoa duyệt	Sau khi sinh viên đã đăng kí xong thì khoa sẽ tiến hành xét duyệt và cập nhật nếu cần thiết. Nếu được chấp nhận thì danh sách đăng kí sẽ được chuyển lên phòng đào tạo nếu không được chấp nhận thì sẽ thông báo cho sinh viên để đăng kí lại.	Văn phòng khoa
10. Phòng đào tạo duyệt	Phòng đào tạo tiến hành duyệt và cập nhật nếu cần thiết.	Văn phòng khoa

Sơ đồ phân rã chức năng



Đặc tả chức năng

Đặc tả chức năng Đăng kí học phần

1. Thông tin chung

a. Mục đích

- Chức năng này cho phép sinh viên đăng kí lớp học phần cho các môn học của mình trong học kỳ.

b. Ngữ cảnh thực hiện

- Chức năng này thực hiện sau chức năng “Xét duyệt môn học”.
- Thực hiện trong thời gian sinh viên tiến hành đăng kí học.

2. Thông tin chi tiết

a. Điều kiện thực hiện

- Người sử dụng phải có quyền truy cập vào chức năng này.

b. Trình tự thực hiện

- Khi sinh viên đăng nhập thì sẽ xác định được Mã sinh viên từ đó xác định được các môn học có thể được đăng kí của sinh viên.
- Chọn lớp học phân tương ứng với từng môn.
- Kiểm tra các điều kiện với môn học và học phần đã chọn, việc kiểm tra được thực hiện như sau:

- Kiểm tra sĩ số: Với mỗi lớp học phần sẽ có sĩ số tối đa, nếu sinh viên đăng kí lớp học phần này đã bằng sĩ số tối đa đó thì sẽ không được đăng kí học phần này.

- Kiểm tra môn song hành: Từ dữ liệu môn học chương trình sẽ lấy được môn học song hành của môn học cần xét, từ dữ liệu điểm thi xét xem sinh viên đã đăng kí học môn học song hành hay chưa ? Nếu chưa thì sinh viên sẽ không được đăng kí môn học đang xét.

- Kiểm tra lịch trùng: khi sinh viên chọn lớp học phần cho một môn học thì chương trình sẽ tiến hành kiểm tra xem với lớp học phần này thì thời khóa biểu của nó có bị trùng với thời khóa biểu của các môn đăng kí trước nó không? Nếu không sẽ không cho sinh viên đăng kí học phần này. Kiểm tra lịch trùng sẽ thực hiện như sau:

+ Lấy danh sách các lớp học phần mà sinh viên đã đăng kí trong học kì này.

+ So sánh thời khóa biểu của lớp học phần đang đăng kí với thời khóa biểu của từng lớp học phần có trong danh sách ở bước trên. Nếu bị trùng thì sẽ không cho đăng kí.

c. Thông tin chi tiết

Thông tin về môn học:

- Mã môn học
- Tên môn học
- Số tín chỉ
- Môn tiên quyết
- Môn trước
- Môn song hành

Thông tin về lớp học phần

- Lớp thứ
- Sĩ số hiện tại
- Sĩ số tối đa
- Buổi thứ
- Từ tiết
- Đến tiết
- Giáo viên
- Phòng học

Đặc tả chức năng Xét duyệt học phần

1. Thông tin chung

a. Mục đích

- Thêm mới sinh viên vào học một lớp học phần.
- Loại bỏ sinh viên ra khỏi lớp học phần.
- Chuyển sinh viên sang các lớp học phần khác.
- Hủy bỏ một lớp học phần.

b. Ngữ cảnh thực hiện

- Thực hiện khi phòng đào tạo tiến hành sắp xếp lịch học.

2. Thông tin chi tiết

a. Điều kiện thực hiện

- Người sử dụng có quyền thực hiện chức năng này.

b. Trình tự thực hiện

- Đầu tiên người sử dụng nhập các thông tin truy vấn: Hệ học, Khóa, Khoa, Chuyên ngành, Bộ môn, Năm học, Học kỳ, Môn học, Lớp học phần.
- Truy vấn ra danh sách các sinh viên trong lớp học phần trên.

- Tiến hành thêm mới sinh viên vào lớp nếu cần thiết.
- Tích chọn các sinh viên muốn chuyển lớp, hoặc loại bỏ khỏi lớp.

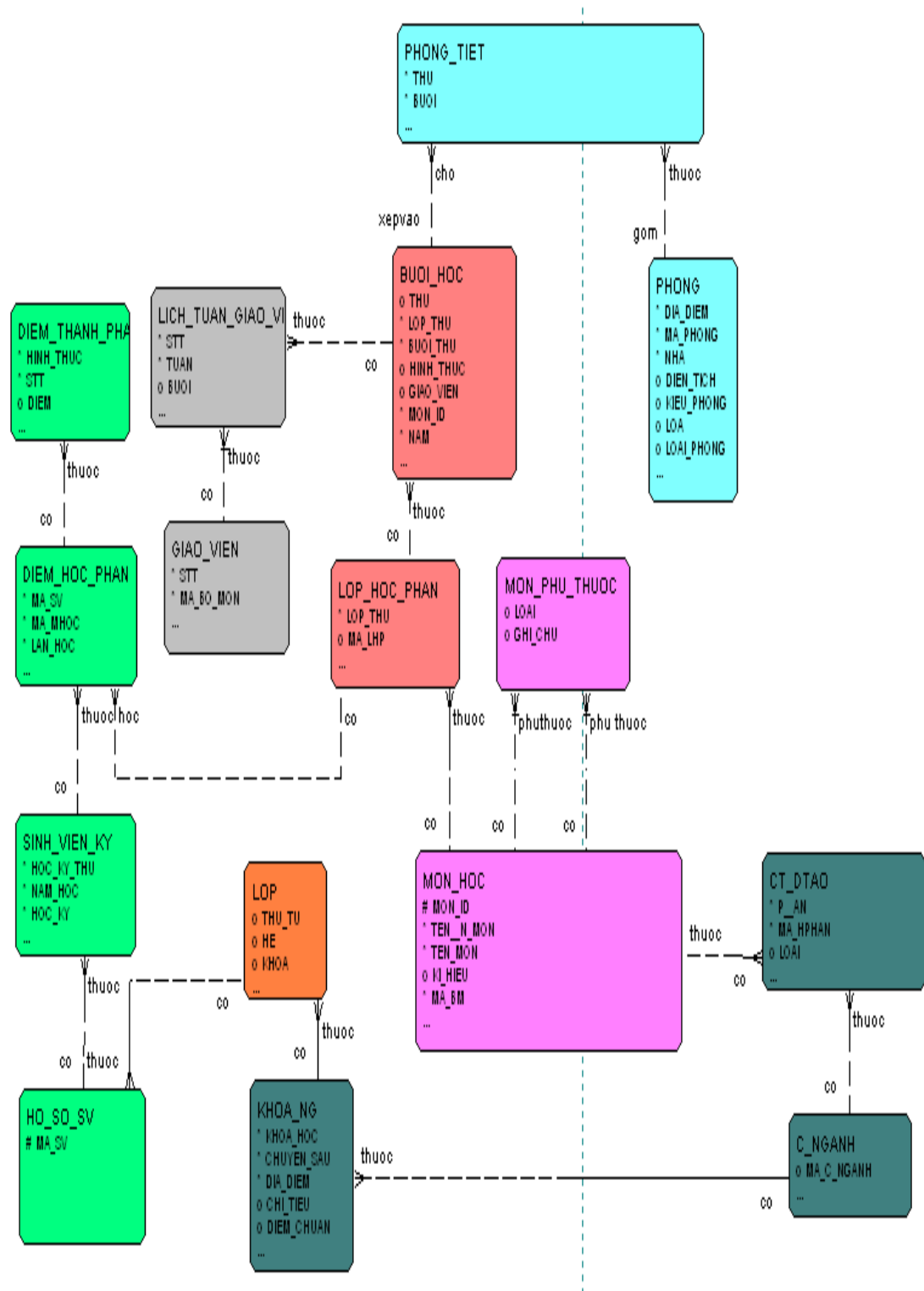
c. Các thông tin chi tiết

Thông tin về sinh viên:

- Mã sinh viên
- Họ và tên
- Tiến độ học
- Số trình bắt buộc tích lũy
- Số trình tích lũy
- Điểm trung bình lần 1
- Điểm trung bình chung
- Trạng thái

Bài 3: Đặc tả thiết kế cơ sở dữ liệu

Sơ đồ quan hệ thực thể



Đặc tả module

Đăng kí học phần

1 Mô tả

a. Mục đích

- Chức năng này cho phép sinh viên đăng kí lớp học phần cho các môn học của mình trong học kì.

b. Các bảng liên quan

+ MON_HOCS
+ LOP_HOC_PHANS
+ BUOI_HOCS
+ HO_SO_SVS
+ SINH_VIEN_KYS
+ DIEM_HOC_PHANS

2. Đặc tả

a. Tham số đầu vào:

- + Mã sinh viên (Được lấy từ User name và Password khi sinh viên đăng nhập hệ thống).
- + Học kỳ thứ - (Tham số này để chọn kỳ đăng kí).

b. Layout chung của chức năng

Module gồm 1 content canvas bao gồm 4 Block:

- Block “Control” dùng để chứa các tham số đầu vào cho chức năng và các button chức năng. Các thuộc tính của block này:
 - + Base table: Unbound block
 - + Layout: Wrap line
 - + Operation: Parameter Form
 - + Canvas: Canvas1
- Block “Diem_hoc_phans” dùng để hiển thị thông tin về các môn học đăng kí. Các thuộc tính của block này:
 - + Base table: DIEM_HOC_PHANS
 - + Layout: 8 row - spread table
 - + Operation: Query only

- + Canvas: Canvas1
- Block “Lop_hoc_phans” dùng để hiển thị thông tin các lớp học phần để sinh viên học. Các thuộc tính của block này:
 - + Base table: Lop_hoc_phans
 - + Title: Thông tin các lớp học phần.
 - + Layout: 4 rows, spread table
 - + Operation: Query only
 - + Canvas: Canvas1
- Block “Buoi_hocs” dùng để hiển thị thông tin về lịch học của các lớp học phần. Các thuộc tính của block này:
 - + Base table: BUOI_HOCS.
 - + Title: Thông tin về lịch học.
 - + Layout: 4 rows, spread sheet
 - + Operation: Query only
 - + Canvas: Canvas1

c. Hoạt động của chức năng

Trên cơ sở các tham số hệ thống sẽ truy vấn ra danh sách các môn học được đăng kí của sinh viên đó trong học kì.

Khi truy vấn ra các môn học thì hệ thống sẽ không cho phép người sử dụng Insert, Update, Delete. Và các thông tin hiển thị ra:

- + Số thứ tự
- + Mã môn.
- + Tên môn.
- + Số tín chỉ.
- + Lần học.
- + Trạng thái đăng kí.
- + Trạng thái học.
- + Trạng thái thi.

Với mỗi một môn học hệ thống sẽ hiển thị ra danh sách các lớp học phần đã mở ra cho môn học đó và người sử dụng không được Insert, Update, Delete. Các thông tin này bao gồm:

- + Số thứ tự
- + Lớp thứ.
- + Sĩ số hiện tại.
- + Sĩ số tối đa.
- + Năm học.
- + Học kỳ.

Khi sinh viên chọn môn học và lớp học phân thích hợp thì sẽ chọn button “Đăng kí” khi đó hệ thống sẽ kiểm tra các điều kiện như: Môn song hành, lịch trùng ... nếu thỏa mãn các điều kiện đó thì đăng kí thành công. Ngoài ra khi sinh viên muốn cập nhật lại bản đăng kí của mình thì chọn button “Danh sách đăng kí”.

Giao diện module

Đăng kí học

Đăng kí học phần

Hệ

Chính qui

Mã SV:

0706869

Họ và tên :

Vũ Xuân Sơn

Khoa :

Cơ khí- Ngành: Máy động lực - K48

Học kỳ thứ đăng ký:

1

Thông tin môn học đăng kí

STT	Mã môn	Tên môn	Số TC	TT Đ.Kí	TT học
1	031	Triết học F2	3	C	C
2	038	Pháp luật đại cương	4	C	C
3	050	Thế dục F1	1	C	C
4	000	Đại số	3	C	C
5	004	Giải tích I (F1)	4	C	C
6	005	Giải tích I (F2)	4	C	C
7	021	Ngoại ngữ F1 (A)	5	C	C
8	030	Triết học F1	6	C	C

Thông tin về lớp học phần

Tên môn	Lớp thứ	Hình thức	Sĩ số HT	Sĩ số TB	Giáo viên
Ngoại ngữ F1	1	LT	10	100	
Ngoại ngữ F1	2		0	100	

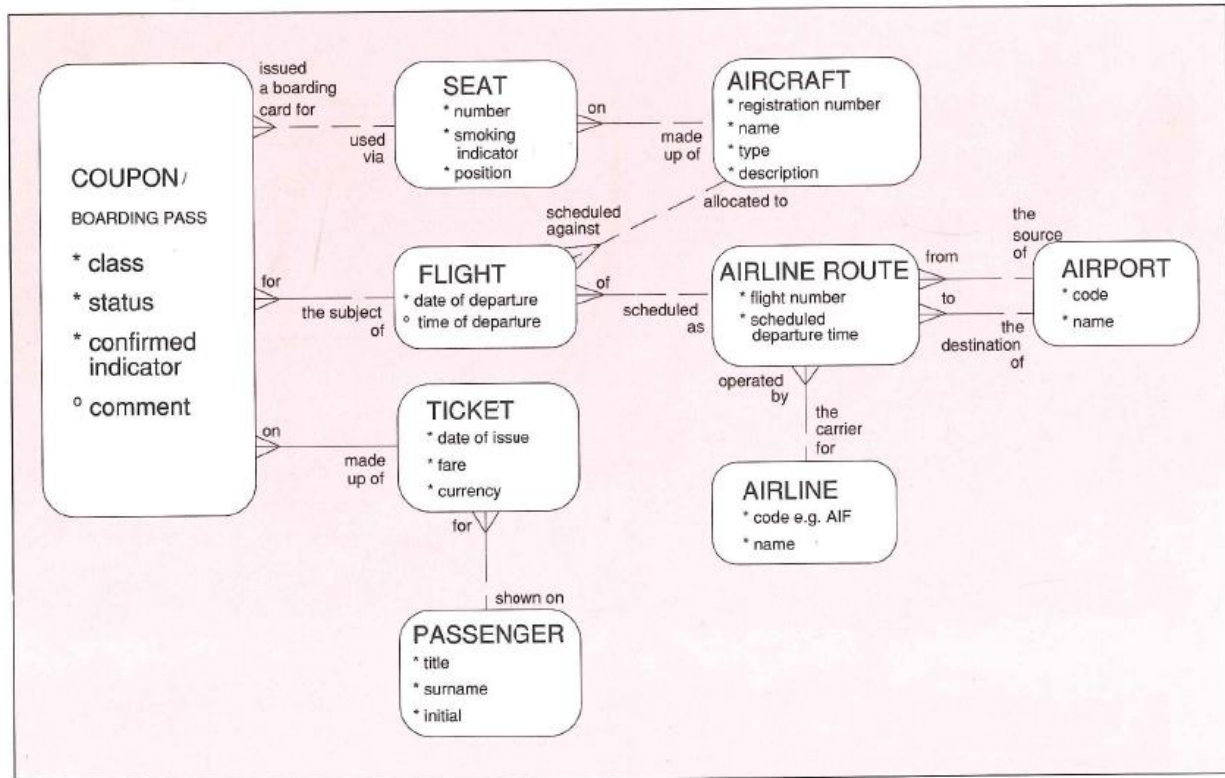
Thông tin buổi học

Hình thức	Buổi thứ	Thứ	Buổi	Phòng	Số tiết	Tiết BD	Ngày bắt đầu	Ngày kết thúc
LT	1	2	8	202A4	3	2	09-JAN-2007	02-JAN-2008

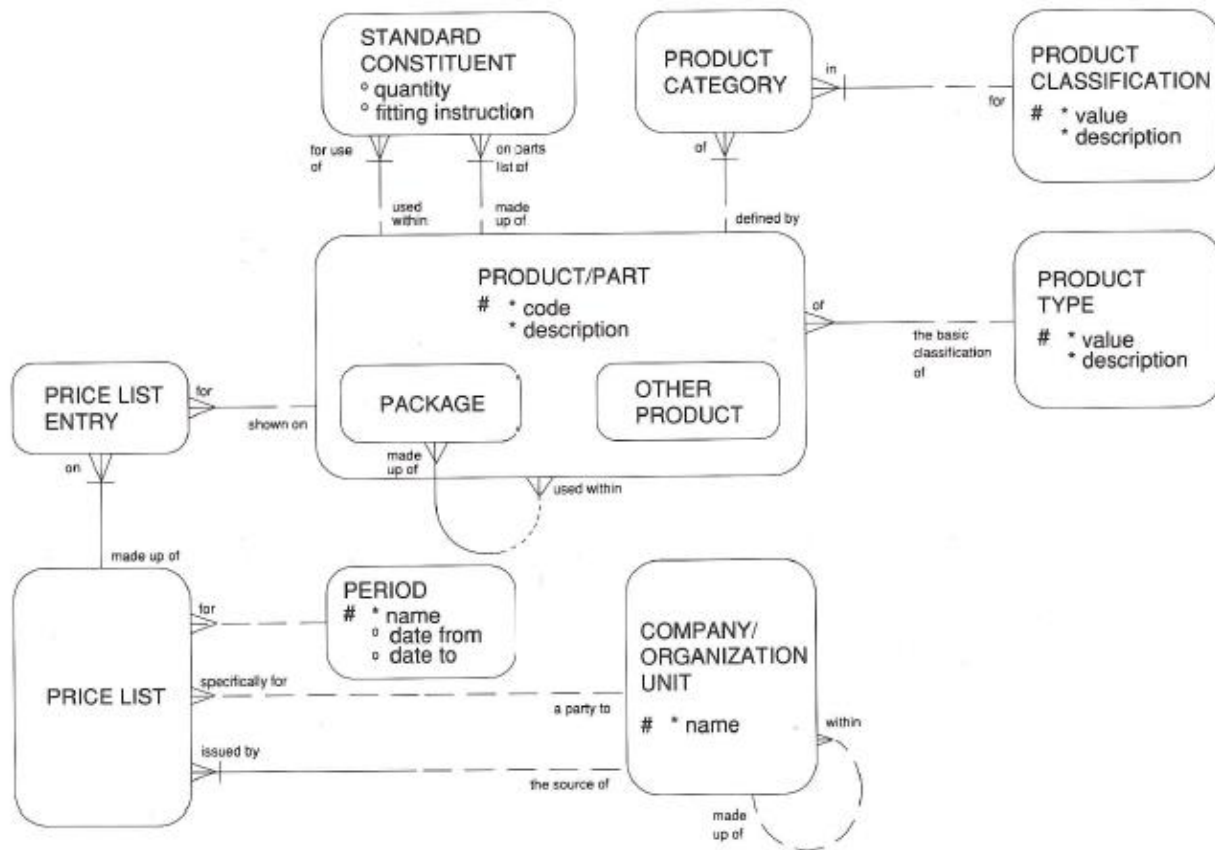
Đăng kí

Danh sách môn đăng kí

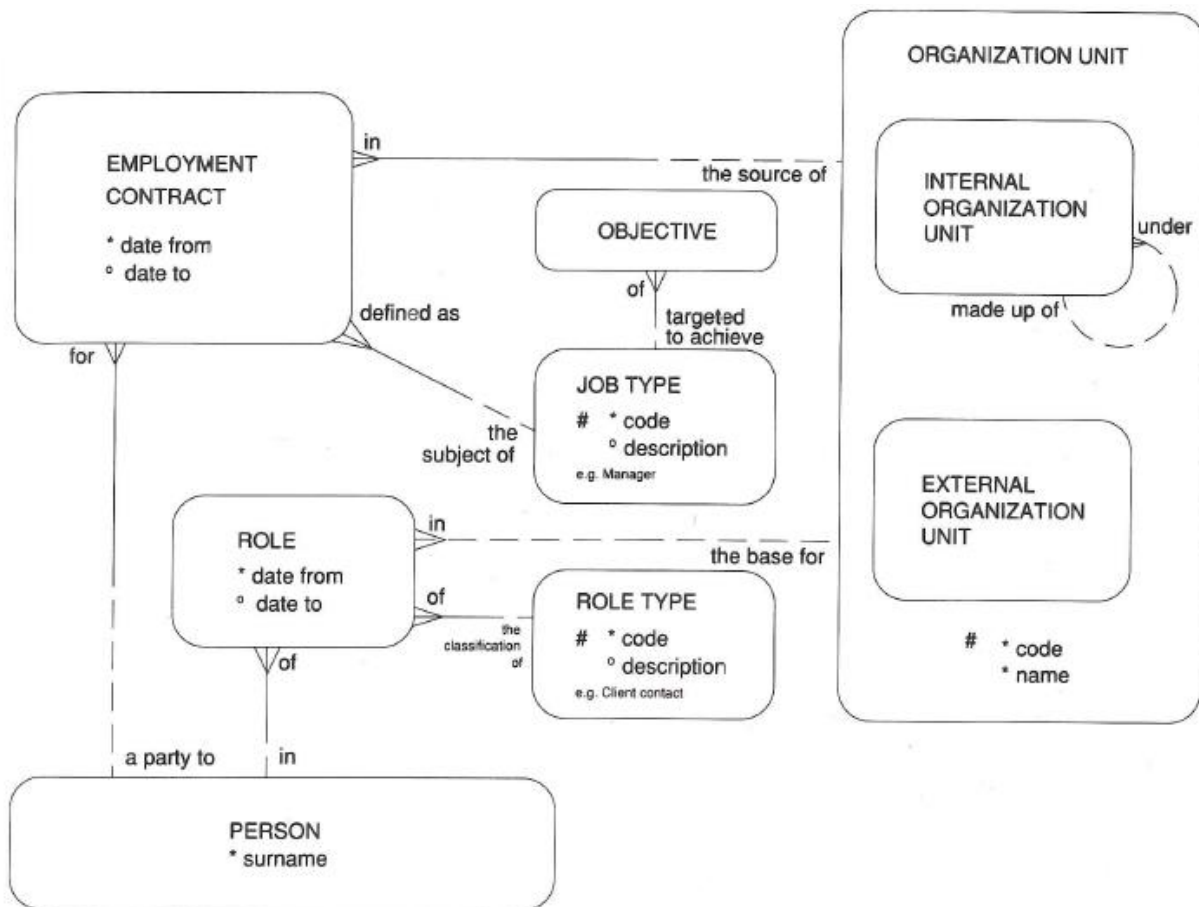
Thiết kế cơ sở dữ liệu cho hệ thống bán vé máy bay



Thiết kế cơ sở dữ liệu cho Hệ thống Quản lý hàng điện tử



Thiết kế cơ sở dữ liệu cho Hệ thống Quản lý nhân sự



Bài 4: Đặc tả hình thức

1. Quan hệ

Tập $\alpha \leftrightarrow \beta$ là tập mọi quan hệ hai ngôi giữa tập α và tập β , có nghĩa là $\alpha \leftrightarrow \beta$ là viết tắt của $P(\alpha \times \beta)$ là tập tất cả các tập con của tích đề các $\alpha \times \beta$, vì mỗi một quan hệ tương ứng với một tập con của tích đề các đó.

Biểu thức $a \rightarrow b$ nghĩa là a là liên quan đến b , mà $a \rightarrow b$ là viết tắt của cặp có thứ tự (a, b) .

Ví dụ

$$\{a \mapsto x, b \mapsto y\} \in (\{a, b\} \leftrightarrow \{x, y\})$$

Quan hệ nhị phân $<$ trên tập các số tự nhiên là kiểu $N \leftrightarrow N$, tức là một phần tử của $N \leftrightarrow N$.

Chúng ta thường viết $x < y$ là viết tắt của ký hiệu $(x, y) \in <$.

Miền xác định dom và miền giá trị ran

$$\begin{aligned} \forall r. (r \in X \leftrightarrow Y \Rightarrow \\ \text{dom } r = \{x \mid x \in X \wedge \exists y. (y \in Y \wedge (x \mapsto y) \in r)\} \wedge \\ \text{ran } r = \{y \mid y \in Y \wedge \exists x. (x \in X \wedge (x \mapsto y) \in r)\}) \end{aligned}$$

Phép hợp quan hệ

$$\begin{aligned} \forall r. (r \in X \leftrightarrow Y \Rightarrow \forall s. (s \in Y \leftrightarrow Z \Rightarrow \\ r ; s = \{x, z \mid x \in X \wedge z \in Z \wedge \exists y. (y \in Y \wedge \\ (x \mapsto y) \in r \wedge (y \mapsto z) \in s)\})) \end{aligned}$$

Ví dụ

$$\begin{aligned} \text{dom}\{a \mapsto x, b \mapsto y, a \mapsto y\} &= \{a, b\} \\ \{a \mapsto x, b \mapsto y, a \mapsto y\} ; \{x \mapsto 1, x \mapsto 3, y \mapsto 2\} \\ &= \{a \mapsto 1, a \mapsto 3, a \mapsto 2, b \mapsto 2\} \end{aligned}$$

\triangleleft là hạn chế miền xác định dom

\triangleleft là hạn chế đối miền xác định

\triangleright là hạn chế miền giá trị ran

\triangleright là hạn chế đối miền giá trị

Chúng được định nghĩa như sau:

$$\begin{aligned} \forall s. (s \in \mathbb{P}X \Rightarrow \forall r. (r \in X \leftrightarrow Y \Rightarrow \\ s \triangleleft r = \{x, y \mid x \in s \wedge (x \mapsto y) \in r\} \wedge \\ s \triangleleft r = \{x, y \mid x \notin s \wedge (x \mapsto y) \in r\})) \\ \forall s. (s \in \mathbb{P}Y \Rightarrow \forall r. (r \in X \leftrightarrow Y \Rightarrow \\ r \triangleright s = \{x, y \mid y \in s \wedge (x \mapsto y) \in r\} \wedge \\ r \triangleright s = \{x, y \mid y \notin s \wedge (x \mapsto y) \in r\})) \end{aligned}$$

Ví dụ

$$\begin{aligned} \{a\} \triangleleft \{a \mapsto x, a \mapsto y, b \mapsto y\} &= \{a \mapsto x, a \mapsto y\} \\ \{a\} \triangleleft \{a \mapsto x, a \mapsto y, b \mapsto y\} &= \{b \mapsto y\} \end{aligned}$$

Nếu chúng ta sử dụng quan hệ để biểu diễn mối quan hệ giữa những cuốn sách được mượn và những người mượn chúng, tức là

$$onLoanTo \in \text{READERID} \leftrightarrow \text{BOOKID}$$

Thì các cuốn sách được mượn bởi một bạn đọc nào đó (rid) sẽ được biểu diễn dạng biểu thức sau

$$\text{ran}(\{rid\} \triangleleft onLoanTo)$$

Ảnh quan hệ

$$\begin{aligned} \forall s. (s \in \mathbb{P}X \Rightarrow \forall r. (r \in X \leftrightarrow Y \Rightarrow \\ r[s] = \{x \mid \exists y. (y \mapsto x \in r \wedge y \in s)\})) \end{aligned}$$

Quan hệ ngược

$$\begin{aligned} \forall r. (r \in X \leftrightarrow Y \Rightarrow \\ r^{-1} = \{x, y \mid y \mapsto x \in r\} \end{aligned}$$

Quan hệ đồng nhất

Cho trước tập S

$$id(S) = \{p \mid \exists x. (x \in S \wedge p = x \mapsto x)\}$$

Ví dụ

$$\{a \mapsto x, a \mapsto y, b \mapsto z\}[\{a\}] = \{x, y\}$$

$$\{a \mapsto x, a \mapsto y, b \mapsto z\}^{-1} = \{x \mapsto a, y \mapsto a, z \mapsto b\}$$

$$id(\{a, b\}) = \{a \mapsto a, b \mapsto b\}$$

Quan hệ (và hàm số) nói chung được sử dụng để lưu giữ dữ liệu và thông thường các quan hệ này cần được cập nhật thông qua các phép toán. Hai phép cập nhật thường dùng là add – thêm một ánh xạ mới hoặc cập nhật một ánh xạ đã có. Để thêm một ánh xạ mới chúng ta có thể sử dụng phép hợp, nhưng để cập nhật chúng ta thường sử dụng phép ghi đè quan hệ.

Ghi đè quan hệ là thường được sử dụng cho các hàm số hơn quan hệ, vì nó giữ tính chất của hàm, trong khi hợp thì không.

Ý tưởng của ghi đè quan hệ là viết đè lên một quan hệ bằng quan hệ khác.

Nó được định nghĩa như sau

$$\forall r, s. (r \in X \leftrightarrow Y \wedge s \in X \leftrightarrow Y \Rightarrow \\ r \triangleleft s = ((\text{dom } s) \triangleleft r) \cup s)$$

Ví dụ

$$\{1 \mapsto a, 2 \mapsto b\} \triangleleft \{2 \mapsto c, 3 \mapsto d\} \\ = \{1 \mapsto a, 2 \mapsto c, 3 \mapsto d\}$$

Bây giờ chúng ta sẽ sử dụng một số kiểu hàm số như: hàm bộ phận, hàm tổng thể, hàm đơn trị và toàn ánh.

Chúng ta bắt đầu xem xét hàm bộ phận, vì nó thường được sử dụng như một kiểu hàm, và vì các kiểu hàm khác là các trường hợp đặc biệt của hàm bộ phận.

Tập $\alpha \twoheadrightarrow \beta$ là tập tất cả các hàm số từ α vào β .

$$\alpha \twoheadrightarrow \beta = \\ \{f \mid f \in \alpha \leftrightarrow \beta \wedge \\ \forall u, v, w. (u \in \alpha \wedge v \in \beta \wedge w \in \beta \Rightarrow \\ (u \mapsto v) \in f \wedge (u \mapsto w) \in f \Rightarrow v = w)\}$$

Ví dụ

Trở lại bài toán hệ thống thư viện, chúng ta thấy đối với mỗi cuốn sách sẽ có nhiều nhất một người mượn và như vậy cách biểu diễn tốt hơn người mượn sẽ là sử dụng hàm số mà gán kết người mượn với các cuốn sách được mượn.

$$onLoanTo \in BOOKID \rightarrow READERID$$

Cho trước bid là cuốn sách được mượn, khi đó người mượn cuốn sách bid đó là

$$onLoanTo(bid)$$

Và tập các cuốn sách được mượn bởi bạn đọc rid là

$$onLoanTo^{-1}[\{rid\}]$$

Chúng ta cũng có thể mô hình bạn đọc rid mượn cuốn sách bid bằng cách cập nhật quan hệ $onLoanTo$ như sau

$$onLoanTo \leftarrow \{bid \mapsto rid\}.$$

Hàm tổng thể là hàm số mà gán giá trị cho mỗi phần tử trên miền xác định

$$S \rightarrow T = \{f \mid f \in S \rightarrow T \wedge \text{dom}(f) = S\}$$

Hàm đơn trị là hàm số mà quan hệ ngược của nó cũng là hàm số

$$S \rightharpoonup T = \{f \mid f \in S \rightarrow T \wedge f^{-1} \in T \rightarrow S\}$$

Hàm toàn ánh là hàm mà ánh xạ từ một giá trị nào đó đến mọi phần tử của miền giá trị.

$$S \twoheadrightarrow T = \{f \mid f \in S \rightarrow T \wedge \text{ran}(f) = S\}$$

Chúng cũng có thể được kết hợp lại như sau

$$\begin{aligned} S \rightharpoonup T &= (S \rightarrow T) \cap (S \twoheadrightarrow T) \\ S \twoheadrightarrow T &= (S \rightarrow T) \cap (S \rightharpoonup T) \\ S \twoheadrightarrow T &= (S \twoheadrightarrow T) \cap (S \rightharpoonup T) \end{aligned}$$

Các dãy thường được sử dụng để biểu diễn dữ liệu mà được sắp thứ tự, chẳng hạn như danh sách, ngăn xếp, hàng đợi.

$$\text{seq } X = \{s \mid s \in \mathbb{N} \rightarrow X \wedge \text{dom } s = 1 \dots \text{card}(s)\}$$

Chúng ta sử dụng $n_1 \dots n_2$, trong đó n_1, n_2 là các số nguyên, để ký hiệu tập các số nguyên nằm giữa hai số nguyên đó, kể cả hai đầu mút.

$$\{n \mid n \in \mathbb{N} \wedge n_1 \leq n \wedge n \leq n_2\}$$

Như vậy, $1 \dots 4$ là viết tắt của tập $\{1, 2, 3, 4\}$

Ví dụ

$$\text{seq}\{a, b\} = \{\{\}, \{1 \mapsto a\}, \{1 \mapsto b\}, \\ \{1 \mapsto a, 2 \mapsto a\}, \{1 \mapsto a, 2 \mapsto b\}, \dots\}$$

Dãy $\{1 \mapsto a, 2 \mapsto b\}$ được viết tắt là $[a, b]$.

Dãy rỗng được ký hiệu là $[\]$.

Phép nối hai dãy được biểu diễn bởi \frown

Ví dụ

$$[a, b] \frown [b, c] = [a, b, b, c]$$

Nếu S là một danh sách, thì $\text{first}(S)$ là phần tử đầu của danh sách và $\text{tail}(S)$ là danh sách còn lại từ S nếu bỏ đi phần tử đầu.

Như vậy đối với mọi danh sách S khác rỗng

$$S = [\text{first}(S)] \frown \text{tail}(S)$$

2. Phương pháp hình thức thiết kế phần mềm

Các khía cạnh của phương pháp thiết kế B-method:

- Ký hiệu để đặc tả phần mềm
- Kỹ thuật để thiết kế phần mềm
- Công cụ hỗ trợ phát triển phần mềm

B method là kết quả của các công trình của Jean-Raymond Abrial, người phát triển nó với đội ngũ nghiên cứu ở Sunbury vào cuối những năm 1980. Ông ta cũng là một trong những người sáng tạo ra ngôn ngữ đặc tả Z.

Để đọc đầy đủ về B method và lý thuyết của nó, có thể sử dụng cuốn sách của Abrial: The B method (nhà xuất bản Cambridge University Press).

B method dùng để đặc tả, thiết kế và lập trình các hệ thống phần mềm.

Các đặc trưng chính gồm:

- Abstract Machine, dữ liệu và các thao tác
- Đặc tả dữ liệu, đặc tả các thao tác
- Tiếp tục làm mịn để cài đặt, các kỹ thuật làm mịn
- Thư viện, tái sử dụng, sinh code
- Chứng minh
- B-toolkit

Nội dung

- Đặc tả phần mềm dùng các ký hiệu và Abstract machine
 - Tin tưởng vào tính đúng đắn của máy
 - Xây dựng máy lớn từ các máy nhỏ
 - Tạo nên phần mềm từ các đặc tả của nó
 - Sử dụng khái niệm làm mịn để liên kết một máy với cài đặt của nó
- **Pro B:** hỗ trợ và kiểm tra mô hình cho B.

Khái niệm máy trừu tượng

Một Abstract Machine bao gồm:

1. Tên máy (có thể với các tham số) được viết bởi mệnh đề MACHINE
2. State – trạng thái
 - VARIABLE: danh sách các tên ký hiệu các thành phần của trạng thái
 - INVARIANT: mệnh đề logic tạo nên các qui tắc tĩnh của máy
 - INITIALISATION: tiền chương trình chỉ ra máy được khởi tạo như thế nào
3. Các thao tác
 - OPERATIONS: Danh sách các thao tác, mà đối với mỗi thao tác, định nghĩa đầu vào, đầu ra, tác động đến các biến trạng thái.

3. Các ví dụ

3.1 Đặc tả và thiết kế ngăn xếp

MACHINE *Stack* (*max* , *CHAR*)
CONSTRAINTS $max \in \mathbb{N}_1$
VARIABLES *stack*
INVARIANT
 $stack \in seq (CHAR) \wedge size (stack) \leq max$
INITIALISATION $stack := []$

OPERATIONS

push (*nn*) $\hat{=}$
 PRE $size (stack) < max \wedge nn \in CHAR$
 THEN $stack := nn \rightarrow stack$
 END ;
pop $\hat{=}$
 PRE $size (stack) > 0$
 THEN $stack := tail (stack)$
 END ;
 $nn \leftarrow \mathbf{top}$ $\hat{=}$
 PRE $size (stack) > 0$
 THEN $nn := first (stack)$
 END

END

3.2 Máy bán hàng

Sau đây chúng ta sẽ đặc tả một Hệ thống hóa đơn cho khách hàng trong môi trường mua thương mại.

Mô tả hệ thống hóa đơn

1. Một khách hàng (client) được lưu lại trong hệ thống cùng với loại khách (category) của anh ta và cũng với phép chi tối đa cho phép. Đối với mỗi loại khách ở đây tương ứng có giảm giá áp dụng cho các hóa đơn tương ứng.
2. Một sản phẩm Product được lưu lại cùng với giá của nó, trạng thái của nó (sẵn sàng hay đã bán hết) và các sản phẩm thay thế có thể, mà là sản phẩm khác mà đảm bảo chưa bán hết.
3. Một hóa đơn liên quan đến khách hàng, mà xuất cho người đó. Hóa đơn cũng có giảm giá được áp dụng cho tổng giá trị của hóa đơn. Cuối cùng nó được đặc trưng bởi tổng tối đa

số tiền mà được chi cho nó. Hai thuộc tính cuối được lấy từ các thuộc tính tương tự của khách hàng.

4. Mỗi dòng của hóa đơn tương ứng với một sản phẩm nào đó, số lượng và giá đơn vị của sản phẩm đó. Thuộc tính cuối được lấy từ thuộc tính giá của sản phẩm.

Các thao tác của Hệ thống hóa đơn

1. Tạo và sửa khách hàng - client
2. Tạo và sửa sản phẩm – product
3. Tạo và xóa hóa đơn – Invoice
4. Thêm dòng mới vào hóa đơn

Các yêu cầu không hình thức

R1. Mọi sản phẩm đã bán hết không là một phần trong hóa đơn

R2. Nếu có tồn tại sản phẩm thay thế cho sản phẩm đã bán hết, thì hệ thống cần thay thế nó trong hóa đơn sản phẩm đó bởi sản phẩm thay thế.

R3. Không có hai dòng khác nhau trong cùng một hóa đơn có thể tương ứng với cùng một sản phẩm.

R4. Không có hóa đơn nào được tạo cho khách hàng loại dubious.

R5. Tổng tiền đã được giảm giá cho một hóa đơn không được lớn hơn số tiền tối đa cho phép chi trả cho hóa đơn đó

R6. Loại khách hàng Friend được giảm 20% giá, trong khi các loại khách khác không được giảm giá.

Xử lý lỗi

Hệ thống cần tạo ra thông báo lỗi khi nhập một sản phẩm mới vào hóa đơn:

1. Sản phẩm đó có thể đã được bán hết và có thể không có sản phẩm thay thế (R1, R2)
2. Sản phẩm hoặc thay thế của nó có thể đã có trong hóa đơn, và không thể có nhiều hơn một dòng ứng với một sản phẩm trong hóa đơn (R2, R3).
3. Số tiền tối đa cho phép trong hóa đơn có thể đạt được khi nhập sản phẩm mới (R2, R5, R6).

Máy client

Máy client đóng gói client và đưa vào hai tập với một hằng số sau:

1. CLIENT là tập dành riêng ký hiệu mọi khách hàng có thể
2. CATEGORY dành cho các loại khách hàng khác nhau.
3. *discount* là hàm liên kết mỗi loại khách hàng với số phần trăm tương ứng được giảm giá cho tổng giá trị trong mọi hóa đơn của họ

State of the *Client* Machine

MACHINE

Client

SETS

CLIENT;

CATEGORY =

{*friend*, *dubious*, *normal*}

CONSTANTS

discount

PROPERTIES

$discount \in CATEGORY \rightarrow (0..100) \wedge$

$discount = \{friend \mapsto 80,$
 $dubious \mapsto 100,$
 $normal \mapsto 100\}$

VARIABLES

client, *category*, *allowance*

INVARIANT

$client \subseteq CLIENT \wedge$

$category \in client \rightarrow CATEGORY \wedge$

$allowance \in client \rightarrow NAT$

INITIALISATION

$client, category, allowance := \emptyset, \emptyset, \emptyset$

The *create_client* Operation

create_client has an input parameter $a : NAT$ representing the allowance given to the newly registered client, who is given *normal* as his category.

```
 $c \leftarrow create\_client(a) \hat{=}$   
PRE  
   $a \in NAT \wedge client \neq CLIENT$   
THEN  
  ANY  
     $cc$   
  WHERE  
     $cc \in CLIENT - client$   
  THEN  
     $client := client \cup \{cc\} \parallel category(cc) := normal \parallel$   
     $allowance(cc) := a \parallel c := cc$   
  END  
END
```

More Operations

```
 $c \leftarrow read\_client \hat{=}$   
PRE  
   $client \neq \emptyset$   
THEN  
   $c := client$   
END
```

```
 $modify\_category(c, k) \hat{=}$   
PRE  
   $c \in client \wedge$   
   $k \in CATEGORY$   
THEN  
   $category(c) := k$   
END
```

```
 $modify\_allowance(c, a) \hat{=} \dots$ 
```

The Product Machine

The *Product* machine is used to encapsulate the product, and introduce its attributes.

MACHINE

Product

SETS

PRODUCT:

$$STATUS = \{available, sold_out\}$$

VARIABLES

product, price,
status, substitute

INVARIANT

$$product \in PRODUCT \wedge$$
$$price \in product \rightarrow NAT \wedge$$
$$status \in product \rightarrow STATUS \wedge$$

substitute ∈

$$product \leftrightarrow status^{-1}[\{available\}]$$

INITIALISATION

```
product, price, status, substitute :=
    0, 0, 0, 0
```

The *create_product* Operation

Function: to create a product. Parameter c is supposed to denote the price of the product. The status of the new product is supposed to be **available**.

$$p \leftarrow \text{create_product}(c) \hat{=}$$

PRE

$$c \in NAT \wedge product \neq PRODUCT$$

THEN

ANY

pp

WHERE

pp ∈ *PRODUCT* – product

THEN

$$price(pp) := c \parallel status(pp) := available \parallel$$
$$product := product \cup \{pp\} \parallel p := pp$$

END

END

More Operations

$make_unavailable(p) \hat{=}$ PRE $p \in product$ THEN $status(p) := sold_out \parallel$ $substitute := substitute \bowtie \{p\}$ END	$assign_substitute(p, q) \hat{=}$ PRE $p \in product \wedge$ $q \in product \wedge$ $status(q) = available$ THEN $substitute(p) := q$ END
	$modify_price(p, c) \hat{=} \dots$ $p \leftarrow read_product \hat{=}$

The *Invoice* Machine

The *Invoice* machine

- **uses** *Client* and *Product* (because it needs to access some of the variables of these machines).
- needs two **sets** *INVOICE* and *LINE* for denoting all the possible invoices and lines.

The *Invoice* Machine (con'd)

MACHINE

Invoice

USES

Client, Product

SETS

INVOICE ;

LINE

VARIABLES

invoice, customer, percentage,
allowed, total, line, origin,
article, quantity, unit_cost

INVARIANT

$invoice \subseteq INVOICE \wedge$
 $customer \in invoice \rightarrow client \wedge$
 $percentage \in invoice \rightarrow (0..100) \wedge$
 $allowed \in invoice \rightarrow NAT \wedge$
 $total \in invoice \rightarrow NAT \wedge$
 $ran(total \otimes allowed) \subseteq leq \wedge$
 $line \subseteq LINE \wedge origin \in line \rightarrow invoice \wedge$
 $article \in line \rightarrow product \wedge$
 $quantity \in line \rightarrow NAT \wedge$
 $unit_cost \in line \rightarrow NAT \wedge$
 $origin \otimes article \in line \mapsto invoice \times product$

where $(f \otimes g)(x) \hat{=} (f(x), g(x))$, all the variables are set to \emptyset initially,
and \mapsto denotes partial injective mapping

Creating an Invoice

This operation is designed to create an invoice for a given non-dubious client ((R4)).

```

inv  $\leftarrow$  create_invoice_header(c)  $\hat{=}$ 
PRE
     $c \in client \wedge category(c) \neq dubious \wedge invoice \neq INVOICE$ 
THEN
    ANY j
    WHERE
         $j \in INVOICE - invoice$ 
    THEN
         $invoice := invoice \cup \{j\} \parallel customer(j) := c \parallel$ 
         $percentage(j) := discount(category(c)) \parallel$ 
         $allowed(j) := allowance(c) \parallel inv := j$ 
    END
END

```


Adding a Line

It adds a line to an invoice when the product has not appeared yet.

```
 $l \leftarrow \text{new\_line}(i, p) \hat{=}$   
PRE  
     $i \in \text{invoice} \wedge p \in \text{product} \wedge \text{status}(p) = \text{available} \wedge$   
     $(i, p) \notin \text{ran}(\text{origin} \otimes \text{article}) \wedge$   
     $\text{line} \neq \text{LINE}$   
THEN  
    ANY  $m$   
    WHERE  $m \in \text{LINE} - \text{line}$   
    THEN  
         $l := m \parallel \text{line} := \text{line} \cup \{m\} \parallel$   
         $\text{origin}(m) := i \parallel \text{article}(m) := p \parallel$   
         $\text{quantity}(m) := 0 \parallel \text{unit\_cost}(m) := \text{price}(p)$   
    END  
END
```

Obtaining a Line

The operation is to obtain the line of an available product, which is supposed to appear already in a certain line of the given invoice.

```
 $l \leftarrow \text{the\_line}(i, p) \hat{=}$   
PRE  
     $i \in \text{invoice} \wedge$   
     $p \in \text{product} \wedge$   
     $\text{status}(p) = \text{available} \wedge$   
     $(i, p) \in \text{ran}(\text{origin} \otimes \text{article})$   
THEN  
     $l := (\text{origin} \otimes \text{article})^{-1}(i, p)$   
END
```

Incrementing a Line

The operation is for incrementing a line, corresponding to an available product, with an extra quantity.

$increment(l, q) \hat{=}$

PRE

$l \in line \wedge$

$q \in NAT \wedge$

$status(article(l)) = available \wedge$

$quantity(l) + q \in NAT \wedge$

$total(origin(l)) +$

$(q \times unit_cost(l) \times percentage(origin(l))/100)$

$\leq allowed(origin(l))$

THEN

$quantity(l) := quantity(l) + q \parallel$

$total(origin(l)) := total(origin(l)) +$

$(q \times unit_cost(l) \times percentage(origin(l))/100)$

Removing All Lines

$remove_all_lines(i) \triangleq$

PRE

$i \in invoice$

THEN

$line := line - origin^{-1}[\{i\}] \parallel$
 $origin := (origin^{-1}[\{i\}] \triangleleft origin) \parallel$
 $article := (origin^{-1}[\{i\}] \triangleleft article) \parallel$
 $quantity := (origin^{-1}[\{i\}] \triangleleft quantity) \parallel$
 $unit_cost := (origin^{-1}[\{i\}] \triangleleft unit_cost)$

where for a function f and a set S we define

$\begin{aligned} \text{dom}(S \triangleleft f) &\triangleq \text{dom}(f) - S \\ (S \triangleleft f)(x) &\triangleq f(x) \end{aligned}$
--

Removing an Invoice

The operation *remove_invoice_header* is used to remove an invoice header.

```
remove_invoice_header(i)  $\hat{=}$   
PRE  
     $i \in \text{invoice} \wedge$   
     $i \notin \text{ran}(\text{origin})$   
THEN  
     $\text{invoice} := \text{invoice} - \{i\} \parallel$   
     $\text{customer} := \{i\} \Leftarrow \text{customer} \parallel$   
     $\text{percentage} := \{i\} \Leftarrow \text{percentage} \parallel$   
     $\text{allowed} := \{i\} \Leftarrow \text{allowed} \parallel$   
     $\text{total} := \{i\} \Leftarrow \text{total}$   
END
```

An invoice can be deleted **after removal of all its lines** using the *remove_all_lines* operation, i.e. $i \notin \text{ran}(\text{origin})$.

The *Invoice_System* Machine

The *Invoice_System* machine is just the joining together of all the previous machines. We add a number of operations to help in defining good error reporting.

MACHINE
Invoice_System
EXTENDS
Client, Product, Invoice

OPERATIONS

$b \leftarrow \text{some_client_exists} \hat{=} \dots;$
 $b \leftarrow \text{clients_not_saturated} \hat{=} \dots;$
 $b \leftarrow \text{client_not_dubious}(c) \hat{=} \dots;$
 $b \leftarrow \text{some_product_exists} \hat{=} \dots;$
 $b \leftarrow \text{products_not_saturated} \hat{=} \dots;$
 $b \leftarrow \text{product_available}(p) \hat{=} \dots;$
 $b \leftarrow \text{product_has_substitute}(p) \hat{=} \dots;$
 $b \leftarrow \text{invoices_not_saturated} \hat{=} \dots;$
 $b \leftarrow \text{new_product_in_invoice}(i, p) \hat{=} \dots$

END

4.2 Hệ thống thang máy

Hệ thống đa thang máy

Một hệ thống thang máy được khởi tạo trong một tòa nhà m tầng. Thiết kế logic để di chuyển các thang máy giữa các tầng tương ứng với các qui tắc sau:

R1. Mỗi thang có một tập các nút nhấn, mỗi nút nhấn cho một tầng. Các nút này sáng khi nhấn và buộc thang máy chờ ở tầng tương ứng. Nút này sẽ tắt sáng khi thang đến tầng đó.

R2. Mỗi tầng (trừ tầng thấp nhất và tầng trên cùng) có hai nút nhấn, một nút để yêu cầu đi lên và một nút để yêu cầu đi xuống. Chúng có thể sáng, khi nhấn. Các nút này sẽ tắt khi có một thang máy đi đến tầng đó theo đúng hướng mong muốn hoặc có thang máy đến tầng đó nhưng không có yêu cầu tiếp theo nào bên trong. Trong trường hợp sau, nếu cả hai nút đều sáng, thì chỉ có một nút tắt đi.

R3. Khi một thang máy không có yêu cầu dịch vụ, nó cần ở lại bên cuối cùng của nó với cửa được đóng và chờ yêu cầu tiếp theo.

R4. Mọi yêu cầu thang máy từ tầng cần phải cuối cùng được phục vụ

R5. Mọi yêu cầu thang máy từ bên trong thang máy cần phải cuối cùng được phục vụ

R6. Mỗi thang máy có nút khẩn cấp, mà khi nhấn, buộc gửi cảnh báo đến quản trị thang. Thang máy đó sau đó được cho rằng sẽ không cung cấp dịch vụ nữa. Mỗi thang máy có cơ chế tắt trạng thái không phục vụ.

The *Lift* Machine

We introduce two sets *LIFT*, which is a deferred set, and the enumerated set *DIRECTION* in to the *Lift* machine. We also define two constants, *top* and *ground*, yielding the top and ground floors.

MACHINE

Lift

SETS

LIFT ;

DIRECTION = {*up*, *dn*}

CONSTANTS

ground, *top*

PROPERTIES

$ground \in NAT \wedge top \in NAT \wedge top > ground$

DEFINITIONS

$FLOOR \triangleq ground..top$

Các biến

1. Biến *moving* được sử dụng để chỉ ra tập các thang máy đang chuyển động.
2. Với mỗi thang máy *l*, ta ký hiệu tầng tương ứng là: *floor(l)*, là tầng mà được giả thiết ở đó thang máy đang dừng, nếu *l* không chuyển động, hoặc tầng gần nhất mà nó sắp đến.
3. Với mỗi thang máy *l*, ta ký hiệu hướng *dir(l)* được giả thiết là hướng theo đó *l* đang chuyển động, nếu nó là *moving* hoặc hướng mà nó hoặc theo hướng của tầng mà nó định đến tiếp theo, nếu nó không chuyển động.
4. Biến *in* là quan hệ hai ngôi từ *FLOOR* vào *DIRECTION*. Khi $(f, d) \in in$, nó có nghĩa là có ai đó muốn di chuyển đến tầng *f* theo hướng *d*.
5. Biến *out* in là quan hệ hai ngôi từ *LIFT* vào *FLOOR*. Khi $(l, f) \in out$, nó có nghĩa là có ai đó đang ở trong thang máy *l* muốn ra khỏi thang máy *l* ở tầng *f*.

Variables (Cont'd)

VARIABLES

moving, floor, dir, in, out

INVARIANT

$moving \subseteq LIFT \wedge$
 $floor \in LIFT \rightarrow FLOOR \wedge$
 $dir \in LIFT \rightarrow DIRECTION \wedge$
 $in \in FLOOR \leftrightarrow DIRECTION \wedge$
 $out \in LIFT \leftrightarrow FLOOR \wedge$
 $(ground \mapsto dn) \notin in \wedge$
 $(top \mapsto up) \notin in \wedge$
 $moving \triangleleft (out \cap floor) = \emptyset \wedge$
 $in \cap ran(moving \triangleleft (floor \otimes dir)) = \emptyset$

INITIALISATION

$in, out, moving := \emptyset, \emptyset, \emptyset \parallel$
 $floor, dir :=$
 $LIFT \times \{ground\}, LIFT \times \{up\}$

Request a Floor

This operation is used to correspond to the event of pressing buttons to request a floor inside a lift.

$Request_Floor(l, f) \triangleq$
PRE
 $l \in LIFT \wedge$
 $f \in FLOOR \wedge$
 $(l \notin moving \Rightarrow (floor(l) \neq f))$
THEN
 $out := out \cup \{l \mapsto f\}$
END

Request a Lift

This operation corresponds to actions of pressing a button to request a lift on a floor.

```
Request_Lift(f, d)  $\hat{=}$   
PRE  
    f  $\in$  FLOOR  $\wedge$   
    d  $\in$  DIRECTION  $\wedge$   
    (f, d)  $\neq$  (ground, dn)  $\wedge$   
    (f, d)  $\neq$  (top, up)  $\wedge$   
    (f, d)  $\notin \text{ran}(\text{moving} \triangleleft (\text{floor} \otimes \text{dir}))$   
THEN  
    in := in  $\cup$  {f  $\mapsto$  d}  
END
```

Các sự kiện điều khiển

Bây giờ chúng ta định nghĩa các sự kiện mà ở đó hệ thống quyết định cho thang máy đang chuyển động, mà đến gần một tầng nào đó, sẽ tiếp tục chuyển động hay là dừng ở tầng đó. Mệnh đề *attracted_up*(*l*) là đúng, khi mà thang máy *l* đang ở hoặc gần đến *floor*(*l*), và ít nhất một trong các điều kiện sau được thỏa mãn

- Một ai đó bên trong thang máy *l* đã thể hiện ý của họ là ra khỏi thang ở một tầng trên tầng *floor*(*l*) theo hướng lên.

$$\text{out}[\{l\}] \cap ((\text{floor}(l) + 1) \dots \text{top}) \neq \emptyset$$

- Có một ai đó đang chờ thang máy ở một tầng trên *floor*(*l*) theo hướng lên của *l*.

$$\text{dom}(\text{in}) \cap ((\text{floor}(l) + 1) \dots \text{top}) \neq \emptyset$$

The DEFINITIONS Clause

Predicate *attracted_dn(l)* is defined in a similar way.

DEFINITIONS

$$\begin{aligned} \text{attracted_up}(l) &\triangleq \\ &(\text{dom}(\text{in}) \cup \text{out}[\{l\}]) \cap ((\text{floor}(l) + 1) \dots \text{top}) \neq \emptyset ; \end{aligned}$$

$$\begin{aligned} \text{attracted_dn}(l) &\triangleq \\ &(\text{dom}(\text{in}) \cup \text{out}[\{l\}]) \cap (\text{ground} \dots (\text{floor}(l) - 1)) \neq \emptyset \end{aligned}$$

Điều khiển chuyển động

can_continue_up(l) và *can_continue_down(l)* được giả thiết là đúng khi một thang máy đang chuyển động *l* không có lý do nào dừng ở tầng *floor(l)* mà nó sắp đến. Rõ ràng điều đó xảy ra khi ba điều kiện sau đồng thời thỏa mãn:

1. Không ai muốn ra từ thang máy *l* ở tầng *floor(l)*.

$$(l \mapsto \text{floor}(l)) \notin \text{out}$$

2. Không ai muốn vào thang máy từ tầng *floor(l)* theo hướng *dir(l)*.

$$(\text{floor}(l) \mapsto \text{dir}(l)) \notin \text{in}$$

3. Thang máy *l* là vẫn được lôi kéo đi theo hướng *dir(l)*

Control of Movement

$can_continue_up(l)$ and $can_continue_down(l)$ are supposed to hold when a moving lift l has no reason to stop at $floor(l)$, where it is about to arrive. Obviously, this is when the following three conditions hold *simultaneously*:

(1) nobody wants to get out from l at $floor(l)$

$$(l \mapsto floor(l)) \notin out$$

(2) nobody wants to get in from $floor(l)$ to travel in $dir(l)$

$$(floor(l) \mapsto dir(l)) \notin in$$

(3) lift l is still attracted in the $dir(l)$ direction.

Continue to Move Up

$$\begin{aligned} can_continue_up(l) \hat{=} & \\ & (l \mapsto floor(l)) \notin out \wedge \\ & (floor(l) \mapsto dir(l)) \notin in \wedge \\ & attracted_up(l) ; \end{aligned}$$

$$\begin{aligned} can_continue_dn(l) \hat{=} & \\ & (l \mapsto floor(l)) \notin out \wedge \\ & (floor(l) \mapsto dir(l)) \notin in \wedge \\ & attracted_dn(l) ; \end{aligned}$$

Lift Control Operations

Continue_up(l) ≐

PRE

$l \in moving \wedge$
 $dir(l) = up \wedge$
 $can_continue_up(l)$

THEN

$floor(l) := floor(l) + 1$

END ;

Stop_up(l) ≐

PRE

$l \in moving \wedge$
 $dir(l) = up \wedge$
 $\neg can_continue_up(l)$

THEN

$moving := moving - \{l\} \parallel$
 $out := out - \{l \mapsto floor(l)\} \parallel$
 $in := in - \{floor(l) \mapsto dir(l)\}$

END

Departure Operations

The decision for the departure of a lift from a floor in a certain direction or for the change of direction of a lift: The key idea is that a lift gives the priority to continuing its travel in the direction it was travelling when it stopped at the floor. If the lift has no reason to travel in the same direction then it is free to change to the opposite direction.

Depart_up(l) ≐

PRE

$l \in LIFT - moving \wedge$
 $dir(l) = up \wedge$
 $attracted_up(l)$

THEN

$moving := moving \cup \{l\} \parallel$
 $floor(l) := floor(l) + 1$

END

Change_up_to_dn(l) ≐

PRE

$l \in LIFT - moving \wedge$
 $dir(l) = up \wedge$
 $\neg attracted_up(l) \wedge$
 $attracted_dn(l)$

THEN

$in := in - \{floor(l) \mapsto dn\} \parallel$
 $dir(l) := dn$

END