

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

LUẬN VĂN THẠC SĨ KHOA HỌC

**NGHIÊN CỨU VÀ ỨNG DỤNG MẪU THIẾT KẾ
TRONG PHƯƠNG PHÁP HƯỚNG ĐỐI TƯỢNG**

**NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ :**

NGÔ THỊ THANH TÂM

Người hướng dẫn khoa học : PGS.TS. ĐẶNG VĂN ĐỨC

HÀ NỘI 2007

MỤC LỤC

DANH MỤC THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT.....	i
DANH MỤC CÁC BẢNG	ii
DANH MỤC CÁC HÌNH VẼ.....	v
MỞ ĐẦU	1
Chương 1. GIỚI THIỆU QUY TRÌNH PHÁT TRIỂN PHẦN MỀM VÀ NGÔN NGỮ MÔ HÌNH HÓA.....	3
1.1 QUY TRÌNH PHÁT TRIỂN PHẦN MỀM.....	3
1.1.1 Định nghĩa.....	3
1.1.2 Phương pháp phát triển phần mềm hướng đối tượng	4
1.1.3 Chu trình phát triển phần mềm xoắn ốc.....	4
1.1.4 Tiến trình phát triển phần mềm RUP.....	6
1.2 NGÔN NGỮ MÔ HÌNH HÓA THỐNG NHẤT - UML	10
1.2.1 Các đặc trưng của UML.....	10
1.2.2 Mô hình khái niệm của UML	11
1.2.3 Kiến trúc hệ thống.....	12
Chương 2. MẪU THIẾT KẾ.....	15
2.1 KHÁI NIỆM CƠ BẢN VỀ MẪU THIẾT KẾ.....	15
2.1.1 Một số định nghĩa	15
2.1.2 Đặc điểm của mẫu thiết kế.....	15
2.1.3 Các yếu tố xác định một mẫu thiết kế.....	15
2.2 MỘT SỐ MẪU THIẾT KẾ	16
2.2.1 Mẫu GRASP	17
2.2.2 Mẫu Gang of Four.....	27
Chương 3. ỨNG DỤNG PHƯƠNG PHÁP HƯỚNG ĐỐI TƯỢNG VÀ MẪU THIẾT KẾ XÂY DỰNG PHẦN MỀM QUẢN LÝ THẺ ĐIỆN THOẠI.....	66

3.1 GIỚI THIỆU BÀI TOÁN	66
3.1.1 Phát biểu bài toán.....	67
3.1.2 Các thành phần của hệ thống	67
3.1.3 Kiến trúc môi trường hệ thống.....	68
3.2 THU THẬP VÀ PHÂN TÍCH YÊU CẦU - MÔ HÌNH USE CASE ..	69
3.2.1 Mục tiêu của hệ thống.....	69
3.2.2 Đặc tả các chức năng hệ thống	69
3.2.3 Nhận biết và mô tả các tác nhân và trường hợp sử dụng.....	71
3.2.4 Biểu đồ Use cases	77
3.2.5 Mô hình hóa nghiệp vụ	77
3.3 THU THẬP VÀ PHÂN TÍCH YÊU CẦU - MÔ HÌNH KHÁI NIỆM	82
3.3.1 Nhận biết các khái niệm (đối tượng)	83
3.3.2 Thuộc tính của các lớp	84
3.3.3 Nhận biết các quan hệ các khái niệm.....	85
3.4 HÀNH VI HỆ THỐNG - CÁC BIỂU ĐỒ TRÌNH TỰ	87
3.4.1 Biểu đồ trình tự hệ thống	87
3.4.2 Giao kèo thao tác của hệ thống.....	88
3.5 THIẾT KẾ HỆ THỐNG	92
3.5.1 Các biểu đồ cộng tác.....	92
3.5.2 Biểu đồ lớp thiết kế.....	99
3.5.3 Thiết kế triển khai	102
3.5.4 Bổ sung thiết kế	106
3.5.5 Mô hình hóa dữ liệu.....	114
3.6 CÀI ĐẶT THIẾT KẾ.....	115
3.6.1 Biểu đồ thành phần	115
3.6.2 Biểu đồ triển khai.....	116
PHẦN KẾT LUẬN.....	118
TÀI LIỆU THAM KHẢO	119

MỞ ĐẦU

Phát triển phần mềm ngày càng trở lên phức tạp. Việc thay đổi giao diện chương trình từ các xâu ký tự sang giao diện đồ họa xu thế sự kiện, từ kiến trúc hệ thống đơn tầng, cơ sở dữ liệu tập trung sang kiến trúc hệ thống đa tầng khách/chủ, cơ sở dữ liệu phân tán, môi trường Internet làm tăng độ phức tạp của hệ thống phần mềm. Thách thức trong 20 năm tới của việc xây dựng hệ thống phần mềm không phải là tốc độ thực hiện chương trình, kinh phí hay sức mạnh của nó mà vấn đề là độ phức tạp. Vậy loại bỏ độ phức tạp bằng cách nào? Các phương pháp tiếp cận hướng cấu trúc, tiếp cận hướng logic, tiếp cận hướng đối tượng và tiếp cận hướng tác tử đều có thể giải quyết vấn đề này nhưng ở những mức độ khác nhau.

Tiếp cận hướng đối tượng đã tỏ ra lợi thế khi lập trình các hệ thống phức tạp. Thực tế cho thấy rằng phát triển phần mềm hướng đối tượng đã và sẽ đem lại phần mềm thương mại chất lượng cao, tin cậy, dễ mở rộng, dễ sử dụng lại, phù hợp với yêu cầu người dùng đang mong đợi. Chúng còn cho khả năng hoàn thành phần mềm đúng thời hạn và với kinh phí thường phù hợp với dự kiến ban đầu.

Với mong muốn tìm hiểu và ứng dụng phương pháp phát triển phần mềm hướng đối tượng để có thể xây dựng các ứng dụng hiệu quả hơn cho ngành bưu điện, học viên đã lựa chọn và tập trung nghiên cứu phương pháp phân tích và thiết kế hướng đối tượng.

Mục đích của luận văn là: nghiên cứu, nắm vững được phương pháp phân tích thiết kế hướng đối tượng, mẫu thiết kế, sử dụng ngôn ngữ mô hình hóa thống nhất UML (*Unified Modeling Language*) và công cụ phần mềm hỗ trợ xây dựng mô hình hệ thống Rational Rose. Đồng thời sử dụng được một số mẫu thiết kế vào công đoạn xây dựng mô hình lớp của quá trình phân tích, thiết kế hệ thống phần mềm theo hướng đối tượng.

Bố cục của luận văn gồm 3 chương, phần mở đầu và phần kết luận.

- Chương 1: Giới thiệu các phương pháp và các quy trình phát triển phần mềm hiện có, tiến trình phát triển phần mềm RUP (*Rational Unified Process*) và ngôn ngữ mô hình hóa thống nhất UML.

- Chương 2: Trình bày khái niệm mẫu thiết kế, ứng dụng mẫu thiết kế và giới thiệu một số mẫu GRASP (*General Responsibility Assignment Software Patterns*) và GoF (*Gang of Four*).

- Chương 3: Trình bày ứng dụng phương pháp phân tích thiết kế hướng đối tượng và một số mẫu thiết kế vào bài toán Quản lý thẻ trả trước tại Bưu điện Thành phố Hà Nội.

Các kết quả của luận án đã bước đầu triển khai ứng dụng thử nghiệm trong hệ thống kinh doanh Thẻ trả trước tại Bưu điện thành phố Hà Nội. Tuy nhiên với thời gian có hạn, luận văn chắc còn nhiều thiếu sót, rất mong nhận được các ý kiến đóng góp của các thầy cô giáo và bạn bè đồng nghiệp.

Chương 1. GIỚI THIỆU QUY TRÌNH PHÁT TRIỂN PHẦN MỀM VÀ NGÔN NGỮ MÔ HÌNH HÓA

1.1 QUY TRÌNH PHÁT TRIỂN PHẦN MỀM

1.1.1 ĐỊNH NGHĨA

Quy trình là phương pháp thực hiện hoặc sản xuất ra sản phẩm.

Quy trình phát triển phần mềm (*Software development/Engineering Process-SEP*) là phương pháp phát triển hay sản xuất ra sản phẩm phần mềm.

Có thể nói quy trình phát triển phần mềm có tính chất quyết định để tạo ra sản phẩm chất lượng tốt với chi phí thấp và năng suất cao.

Thông thường một quy trình bao gồm những yếu tố cơ bản sau:

- Thủ tục
- Hướng dẫn công việc
- Biểu mẫu
- Danh sách kiểm định
- Công cụ hỗ trợ

Với các nhóm công việc chính:

- Đặc tả yêu cầu: chỉ ra những “đòi hỏi” cho cả các yêu cầu chức năng và phi chức năng.
- Phát triển phần mềm: Tạo ra phần mềm thỏa mãn các yêu cầu được chỉ ra trong “Đặc tả yêu cầu”.
- Kiểm thử phần mềm: Để bảo đảm phần mềm sản xuất ra đáp ứng những “đòi hỏi” được chỉ ra trong “Đặc tả yêu cầu”.
- Thay đổi phần mềm: Đáp ứng nhu cầu thay đổi của khách hàng.

Tùy theo mô hình phát triển phần mềm, các nhóm công việc được triển khai theo những cách khác nhau. Để sản xuất cùng một sản phẩm phần mềm

người ta có thể dùng các mô hình khác nhau. Tuy nhiên không phải tất cả các mô hình đều thích hợp cho mọi ứng dụng.

1.1.2 PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

Quan điểm hướng đối tượng hình thành trên cơ sở tiếp cận hướng hệ thống, nó coi hệ thống như thực thể được tổ chức từ các thành phần mà chỉ được xác định khi nó thừa nhận và có quan hệ với các thành phần khác. Phương pháp tách vấn đề đang giải quyết để hiểu chúng ở đây không chỉ dựa trên cơ sở hệ thống làm mà còn dựa trên việc tích hợp hệ thống là cái gì và hệ thống làm gì. Theo cách tiếp cận hướng đối tượng các chức năng hệ thống được biểu diễn thông qua cộng tác của các đối tượng, việc thay đổi, tiến hoá chức năng sẽ không ảnh hưởng đến cấu trúc tĩnh của phần mềm. Sức mạnh của tiếp cận hướng đối tượng là việc tách (chia) và nhập (thống nhất) được thực hiện nhờ tập phong phú các cơ chế tích hợp của chúng; khả năng thống nhất cao nhưng nó đã được tách ra để xây dựng các thực thể phức tạp từ các thực thể đơn giản.

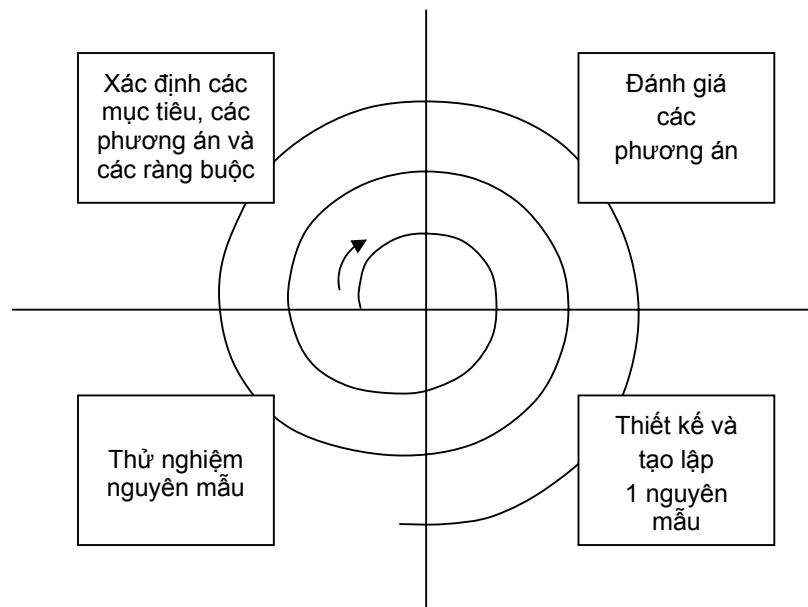
Tiếp cận hướng đối tượng đã tỏ ra lợi thế khi lập trình các hệ thống phức tạp. Những người phát triển phần mềm nhận thấy rằng phát triển phần mềm hướng đối tượng sẽ đem lại phần mềm thương mại chất lượng cao, tin cậy dễ mở rộng và dễ sử dụng lại, phù hợp với yêu cầu người dùng đang mong đợi. Chúng còn cho khả năng hoàn thành phần mềm đúng thời hạn và không vượt quá kinh phí dự kiến ban đầu.

Phương pháp hướng đối tượng ra đời từ những năm 1990 và đến năm 1997 đã được quy chuẩn qua ngôn ngữ mô hình hóa thống nhất UML.

1.1.3 CHU TRÌNH PHÁT TRIỂN PHẦN MỀM XOẺN ỐC

Mọi hệ thống đều phải trải qua sự khởi đầu, triển khai, xây dựng, khai thác, bảo dưỡng và kết thúc, đó chính là vòng đời. Khi quan tâm đến sự triển khai và xây dựng tức là quan tâm đến sự phát triển hệ thống, trong đó khía cạnh

quy trình phát triển phần mềm (còn gọi là chu trình) là sự tiếp nối các thời kỳ trong phát triển hệ thống. Có nhiều loại chu trình phát triển phần mềm khác nhau như chu trình thác nước, chu trình chữ V, chu trình tăng trưởng, chu trình xoắn ốc,... [1]. Trong đó chu trình xoắn ốc được là mô hình tổng quát nhất, tất cả các mô hình khác đều có thể xem là một hiện thực của mô hình tổng quát này, hay cũng có thể xem nó là mô hình tổng hợp các mô hình khác. Đặc biệt, chu trình xoắn ốc được ứng dụng không chỉ trong phát triển phần mềm mà còn trong phát triển phần cứng.



Hình 1.1 Chu trình xoắn ốc

Quy trình xoắn ốc (hình 1.1) hay quy trình lặp có các đặc điểm :

- Tiến trình lặp đi lặp lại một dãy các giai đoạn nhất định.
- Qua mỗi vòng lặp, tạo ra một phiên bản hoàn thiện dần.
- Nhấn mạnh sự khắc phục các nguy cơ (một nguy cơ bắt nguồn từ các sai sót trong sự đặc tả các nhu cầu).

Quy trình xoắn ốc cung cấp một phần hệ thống để khách hàng có thể đưa vào sử dụng trong môi trường hoạt động sản xuất thực sự mà không cần chờ cho

đến khi toàn bộ hệ thống được hoàn thành. Để khách hàng có thể sử dụng, mỗi phiên bản đều phải được thực hiện như một quy trình đầy đủ các công việc từ phân tích yêu cầu với khả năng bổ sung hay thay đổi, thiết kế, hiện thực cho đến kiểm nghiệm và có thể xem như một quy trình (chu trình) con. Các chu trình con có thể sử dụng các mô hình khác nhau (thông thường là mô hình thác nước).

Mục tiêu của phiên bản đầu tiên là phát triển phần lõi và nhóm các chức năng quan trọng. Sau mỗi phiên bản được đưa vào sử dụng, các kết quả đánh giá sẽ được phản hồi và lập kế hoạch cho chu trình con của phiên bản tiếp theo để thực hiện:

- Những thay đổi cho phiên bản trước đó nhằm đáp ứng nhu cầu khách hàng tốt hơn.
- Có thể thêm những chức năng hoặc đặc điểm bổ sung.

Các vòng lặp được tiếp tục cho đến khi xét thấy nguyên mẫu là tốt để có thể chuyển sang sản xuất thực sự được.

Đây chính là mô hình tổng quát nhất, tất cả các mô hình khác đều có thể xem là một hiện thực của mô hình tổng quát này, hay cũng có thể xem nó là mô hình tổng hợp các mô hình khác. Đặc biệt, chu trình xoắn ốc được ứng dụng không chỉ trong phát triển phần mềm mà còn trong phát triển phần cứng.

Quy trình phát triển RUP mà luận văn giới thiệu ở phần tiếp theo chính là một ví dụ điển hình của quy trình này.

1.1.4 TIẾN TRÌNH PHÁT TRIỂN PHẦN MỀM RUP

1.1.4.1 Tổng quan về quy trình phát triển RUP

Một quy trình chuẩn được công nhận trong quá trình phân tích thiết kế, phát triển, thử nghiệm, triển khai chương trình sẽ quyết định chất lượng của chương trình tại thời điểm tiến hành triển khai thử nghiệm.

RUP là một tiến trình phát triển phần mềm do hãng Rational xây dựng, nó cung cấp một nguyên tắc tiếp cận để gán nhiệm vụ và trách nhiệm trong một tổ chức phát triển. Mục tiêu là đảm bảo sản phẩm phần mềm chất lượng cao mà thoả mãn các nhu cầu của người sử dụng, đúng kế hoạch và kinh phí [8].

RUP bắt kịp nhiều phương pháp tốt nhất trong việc phát triển phần mềm hiện đại với mẫu phù hợp với nhiều dự án và tổ chức. Nó mô tả các cách tiếp cận đã được thử nghiệm về phương diện thương mại để triển khai có hiệu quả tới việc phát triển phần mềm cho các nhóm phát triển phần mềm.

RUP cung cấp cho mọi thành viên trong nhóm các hướng dẫn, khuôn mẫu, công cụ hướng dẫn cần thiết cho cả nhóm để tận dụng các bài thực hành tối ưu sau :

i) *Phát triển lặp:* RUP chia quá trình phát triển thành các chu kỳ khác nhau, ở những chu kỳ đầu sẽ lựa chọn phát triển trước những chức năng mẫu chốt, quyết định toàn bộ sự thành công hay thất bại của dự án, mỗi chu kỳ như vậy sẽ sinh ra một phiên bản thi hành được của ứng dụng đang phát triển.

ii) *Quản lý các yêu cầu:* Đảm bảo giải quyết đúng vấn đề gặp phải và xây dựng đúng hệ thống cần xây dựng; quản trị yêu cầu cho phép theo vết được các vấn đề đặt ra từ nhu cầu của người sử dụng hệ thống đến các đặc tính của hệ thống, các chức năng, các vấn đề về phân tích, thiết kế và kịch bản thử nghiệm.

iii) *Sử dụng kiến thức thành phần:* Chia nhỏ hệ thống phần mềm ra các thành phần nhỏ tương đối độc lập nhưng lại có quan hệ với nhau theo những nguyên tắc nhất định.

iv) *Mô hình trực quan phần mềm:* RUP Sử dụng ngôn ngữ chuẩn UML để mô hình hóa toàn bộ hệ thống phần mềm cần phát triển.

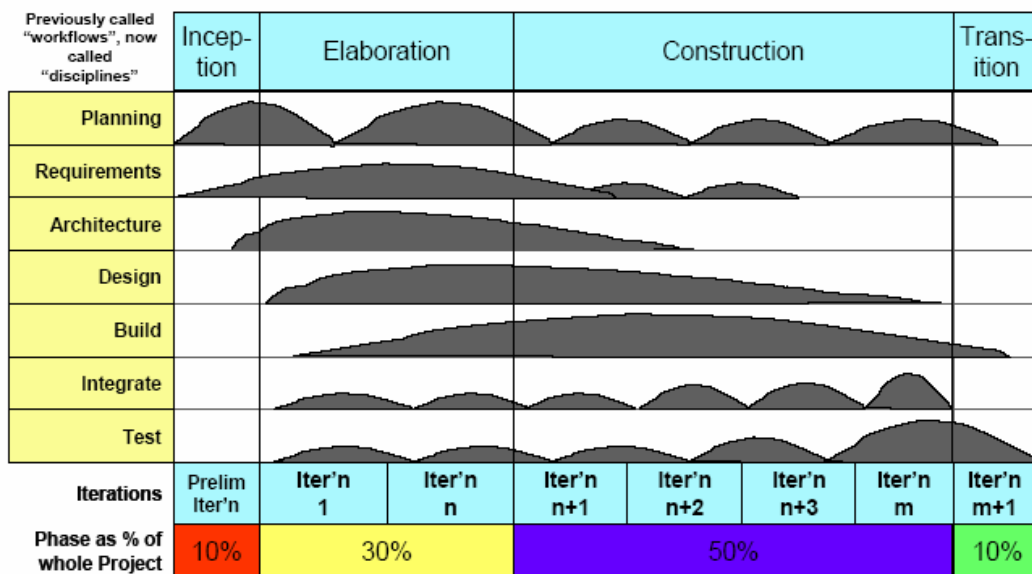
v) *Kiểm tra chất lượng phần mềm:* RUP cho phép việc kiểm tra thử nghiệm được thực hiện ở tất cả các chu kỳ phát triển ứng dụng và kiểm tra trên cả 3 mặt chính: kiểm tra về mặt chức năng ứng dụng (thử nghiệm tất cả các kịch

bản tình huống sử dụng), kiểm tra tốc độ (hiệu năng) và kiểm tra độ tin cậy của ứng dụng.

vi) Điều khiển các thay đổi tới phần mềm : Khả năng quản lý sự thay đổi, duy trì được sự ổn định khi có biến động hay phát hiện sự biến động là rất cần thiết. RUP cho cách tìm ra, kiểm soát và xử lý những biến đổi này. RUP cũng hướng dẫn cách thành lập vùng làm việc an toàn cho mỗi lĩnh vực bằng việc tách rời các biến động từ các bộ phận khác nhau, kiểm soát sự biến động của các chế tác phần mềm. Do đó một nhóm có thể hoạt động như một đơn vị độc lập và có thể tự động tương tác và xây dựng quản lý.

1.1.4.2 Quy trình phát triển phần mềm theo RUP

Tiến trình RUP được mô tả theo hai chiều (hai trục) (hình 1.2): Trục hoành thể hiện thời gian và chỉ ra khía cạnh động của tiến trình. Trục tung biểu diễn khía cạnh tĩnh của tiến trình, mô tả hoạt động, chế tác, nhân viên và luồng công việc.



Hình 1.2 Tiến trình RUP

Quy trình phát triển phần mềm theo RUP gồm 4 pha (pha khởi đầu, pha chi tiết, pha xây dựng và pha chuyển giao) và các giai đoạn công việc mà đội thực hiện dự án cần tuân theo. Kết thúc mỗi pha là mốc được xác định rõ ràng, đó là thời điểm phải đưa ra quyết định quan trọng để đạt được mục tiêu mấu chốt.

Pha khởi đầu (Inception): Trong pha này, nhà phát triển hình thành các ca nghiệp vụ cho hệ thống và phân định phạm vi dự án. Để thực hiện điều này nhà phát triển phải nhận ra được các thực thể ngoài tương tác với hệ thống (tác nhân) và xác định bản chất của tương tác này. Việc này dẫn đến nhận dạng các ca nghiệp vụ bao gồm các tiêu chí thắng lợi, đánh giá rủi ro, dự báo tài nguyên cần thiết và kế hoạch pha của các mốc chính. Kết quả pha này là chúng ta xác định được mục đích của dự án và đưa ra các bước cần thiết để tiếp tục phát triển dự án.

Pha chi tiết (Elaboration): Pha chi tiết bắt đầu bằng phân tích yêu cầu và mô hình hóa lĩnh vực. Mục tiêu của pha này là phân tích vấn đề, lựa chọn và hình thành lên nền tảng kiến trúc, hạn chế nguyên nhân rủi ro của dự án, xác định kế hoạch đầy đủ cho các nhiệm vụ phát triển hệ thống phần mềm. Để đạt được mục đích này các quyết định kiến trúc phải được thực hiện để hiểu toàn bộ hệ thống bao gồm: phạm vi, các chức năng chính, các yêu cầu phi chức năng.

Pha chi tiết là pha quan trọng nhất trong bốn pha. Kết thúc pha này, vấn đề kỹ nghệ phải được xem xét đầy đủ, dự án phải được tính toán kỹ để quyết định có hay không cam kết thực hiện các pha xây dựng và chuyển giao. Các hoạt động pha chi tiết đảm bảo rằng kiến trúc, yêu cầu và kế hoạch là đủ ổn định, các rủi do bị hạn chế, do vậy có thể dự báo giá cả, thời gian để hoàn thành việc phát triển.

Pha xây dựng (Construction): Trong pha này, mọi thành phần còn lại và các đặc trưng ứng dụng được phát triển và tích hợp vào ứng dụng, mọi đặc trưng phải được kiểm thử. Pha xây dựng tập trung vào quản lý tài nguyên và thực hiện các công việc tối ưu giá cả, thời gian và chất lượng. Nhiều dự án có các hoạt động song song có thể đẩy nhanh các kết quả có giá trị. Kiến trúc và kế hoạch có

mối quan hệ chặt chẽ. Nói cách khác chất lượng cao của kiến trúc là một thuận lợi cho xây dựng. Đây là lý do tại sao sự phát triển thăng bằng của kiến trúc và kế hoạch được nhấn mạnh trong pha chi tiết.

Kết quả của pha xây dựng là sản phẩm sẵn sàng đặt trong tay người sử dụng. Nó có thể bao gồm: Sản phẩm phần mềm tích hợp, Tài liệu hướng dẫn sử dụng, Mô tả phiên bản hiện hành.

Pha chuyển giao (Transition): Mục đích của pha này là chuyển giao sản phẩm đến cộng đồng người sử dụng. Một khi sản phẩm đã đến tay người dùng thì nhiệm vụ đòi hỏi ta phải phát triển phiên bản mới, sửa lỗi nếu có hay kết thúc các đặc trưng còn chưa hoàn thành.

Pha này được bắt đầu khi các cơ sở đã tương đối hoàn chỉnh để triển khai đến người dùng cuối. Những yêu cầu chính thức mà một số sản phẩm thử nghiệm của hệ thống được hoàn thiện và đạt yêu cầu chất lượng mà chuyển giao cho người sử dụng sẽ tạo ra những kết quả tích cực cho tất cả các bên.

1.2 NGÔN NGỮ MÔ HÌNH HÓA THỐNG NHẤT - UML

UML là một ngôn ngữ mô hình hóa thống nhất. Nó là ngôn ngữ mô hình hóa chuẩn để thiết kế phần mềm hướng đối tượng, được dùng để đặc tả, trực quan hóa, xây dựng và làm tài liệu cho các hệ thống phần mềm [10].

1.2.1 CÁC ĐẶC TRƯNG CỦA UML

UML còn là ngôn ngữ đồ họa với các tập quy tắc và ngữ nghĩa, nó cho ta biết cách tạo ra và đọc hiểu được một mô hình thiết kế một cách rõ ràng.

UML là một ngôn ngữ làm trực quan. Những điều suy nghĩ và hình dung về một hệ thống cần xây dựng từ các khía cạnh khác nhau được biểu diễn bằng ký hiệu đồ họa và các biểu diễn bằng sơ đồ với các giải thích bằng văn bản đi kèm.

UML là một ngôn ngữ đặc tả có cấu trúc. UML cho phép mô tả mô hình chính xác, không nhập nhằng và hoàn thiện. UML hướng tới đặc tả thiết kế, phân tích và quyết định cài đặt trong quá trình phát triển và triển khai hệ thống phần mềm.

UML là ngôn ngữ để xây dựng. UML không phải là ngôn ngữ lập trình trực quan, nhưng mô hình của nó có thể kết nối trực tiếp với các ngôn ngữ lập trình khác nhau bằng việc ánh xạ mô hình trong UML tới các ngôn ngữ lập trình khác nhau như JAVA, C++ hay các bảng cơ sở dữ liệu (CSDL) quan hệ, CSDL hướng đối tượng.

UML là một ngôn ngữ làm tài liệu. UML hướng tới làm tài liệu kiến trúc hệ thống và các chi tiết của nó. UML cho khả năng biểu diễn yêu cầu, thử nghiệm, mô hình hoá các hoạt động lập kế hoạch và quản lý sản phẩm.

1.2.2 MÔ HÌNH KHÁI NIỆM CỦA UML

Mô hình khái niệm của UML gồm ba vấn đề chính: các phần tử cơ bản để xây dựng mô hình, các quy tắc liên kết và các cơ chế chung được sử dụng cho ngôn ngữ.

Các khối để hình thành mô hình UML bao gồm: Phần tử, Quan hệ và Biểu đồ.

i) Các phần tử trong UML gồm 4 loại:

- Phần tử cấu trúc gồm có: lớp, giao diện, phần tử cộng tác, trường hợp sử dụng (*Use case*), lớp tích cực (*active class*), thành phần và nút (*node*).
- Phần tử hành vi gồm có: tương tác, máy trạng thái.
- Phần tử nhóm chỉ có một phần tử là gói (*package*).
- Chú thích (*annotational*).

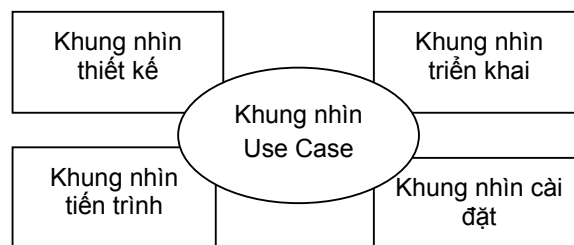
ii) Các quan hệ trong UML bao gồm 4 loại: quan hệ phụ thuộc (*dependency*), quan hệ kết hợp (*association*), quan hệ khái quát hóa (*generalization*) và quan hệ hiện thực hóa (*realization*).

iii) Biểu đồ UML gồm có: biểu đồ trường hợp sử dụng (*User case - UC*), biểu đồ trình tự (*sequence*), biểu đồ cộng tác (*collaboration*), biểu đồ lớp (*class*), biểu đồ chuyển trạng thái (*state transition*), biểu đồ thành phần (*component*) và biểu đồ triển khai (*deployment*).

Chi tiết về các khối để hình thành mô hình UML được mô tả chi tiết trong các tài liệu [2, 7].

1.2.3 KIẾN TRÚC HỆ THỐNG

Kiến trúc phần mềm cho phép nhìn khái quát nhất về hệ thống phần mềm ở các góc độ khác nhau. Kiến trúc của hệ thống phần mềm được mô tả bằng năm loại khung nhìn tương tác với nhau (hình 1.3). Mỗi khung nhìn phản ánh về một khía cạnh của tổ chức và cấu trúc của hệ thống và tập trung vào từng mặt cụ thể giúp cho việc hiểu và sử dụng hệ thống được tốt nhất.



Hình 1.3 Mô hình hoá kiến trúc hệ thống

i) Khung nhìn Use Case đứng trước mọi khung nhìn khác. Nó được hình thành từ giai đoạn phân tích yêu cầu và được sử dụng để điều khiển và thúc đẩy phần việc còn lại của thiết kế. Khung nhìn này mô tả các hành vi hệ thống theo cách nhìn của khách hàng, của nhà phân tích và người kiểm thử. Khung nhìn Use Case chứa các tác nhân: UC, biểu đồ UC, một vài biểu đồ trình tự, biểu đồ

cộng tác và gói. Khung nhìn này tập trung vào mức cao của cái hệ thống sẽ làm, không quan tâm đến hệ thống làm như thế nào.

ii) Khung nhìn thiết kế tập trung vào việc hệ thống cài đặt các hành vi trong Use Case như thế nào. Nó bao gồm các lớp, biểu đồ lớp, biểu đồ đối tượng (khía cạnh tĩnh của khung nhìn), biểu đồ tương tác, biểu đồ biến đổi trạng thái (khía cạnh động của hệ thống) và các gói. Thông thường đội ngũ phát triển phần mềm tiếp cận khung nhìn thiết kế theo hai bước:

- Bước thứ nhất: Nhận ra các lớp phân tích là các lớp độc lập với ngôn ngữ lập trình.

- Bước thứ hai: Chuyển các lớp phân tích sang lớp thiết kế là những lớp phụ thuộc ngôn ngữ lập trình.

Khung nhìn thiết kế tập trung vào cấu trúc logic của hệ thống. Trong khung nhìn này ta sẽ nhận ra các bộ phận hệ thống, khảo sát thông tin và hành vi, khảo sát quan hệ giữa các bộ phận.

iii) Khung nhìn cài đặt hay khung nhìn thành phần gồm các thành phần là mô-đun vật lý hay tệp mã trình để lắp ráp thành hệ thống vật lý. Khung nhìn thành phần bao gồm thành phần, biểu đồ thành phần và gói.

iv) Khung nhìn tiến trình của hệ thống chứa đựng các luồng và tiến trình công việc tạo nên cơ chế hoạt động tương tranh và đồng bộ của hệ thống.

v) Khung nhìn triển khai tập trung vào phân bổ vật lý của các tài nguyên và phân bổ nhiệm vụ giữa các tài nguyên. Khung nhìn này chỉ ra các tiến trình và thiết bị trên mạng và các kết nối vật lý giữa chúng.

Tuy nhiên không phải tất cả các hệ thống đều đòi hỏi đầy đủ các khung nhìn như mô tả trên. Ví dụ: hệ thống trên máy tính riêng lẻ có thể bỏ khung nhìn triển khai, nếu hệ đơn xử lý thì bỏ khung nhìn tiến trình, nếu chương trình nhỏ thì bỏ khung nhìn cài đặt.

Tóm lược: Chương 1 đã trình bày các nội dung nghiên cứu tổng quan về phương pháp và quy trình phát triển phần mềm, ngôn ngữ mô hình hóa UML. Trong đó luận án tập trung nghiên cứu phương pháp phát triển phần mềm hướng đối tượng và tiến trình phát triển phần mềm RUP.

MỞ ĐẦU	1
Chương 1. GIỚI THIỆU QUY TRÌNH PHÁT TRIỂN PHẦN MỀM VÀ NGÔN NGỮ MÔ HÌNH HÓA	3
1.1 Quy trình phát triển phần mềm.....	3
1.1.1 Định nghĩa.....	3
1.1.2 phương pháp phát triển phần mềm HUỚNG ĐỐI TƯỢNG.....	4
1.1.3 CHU trình phát triển phần mềm XOẮN ỐC	4
1.1.4 Tiển trình phát triển phần mềm RUP	6
1.1.4.1 Tổng quan về quy trình phát triển RUP	6
1.1.4.2 Quy trình phát triển phần mềm theo RUP	8
1.2 Ngôn ngữ mô hình hóa thỔng nhẮt - UML	10
1.2.1 Các đẶc trưng cỦa UML.....	10
1.2.2 Mô hình khái niỆm cỦa UML	11
1.2.3 KIẾN TRÚC HỆ THỐNG	12

Hình 1.1 Chu trình xoắn ốc	5
Hình 1.2 Tiến trình RUP	8
Hình 1.3 Mô hình hoá kiến trúc hệ thống	13

Chương 3. ỨNG DỤNG PHƯƠNG PHÁP HƯỚNG ĐỐI TƯỢNG VÀ MẪU THIẾT KẾ XÂY DỰNG PHẦN MỀM QUẢN LÝ THẺ ĐIỆN THOẠI

Chương này trình bày việc ứng dụng quy trình RUP và phương pháp hướng đối tượng và mẫu thiết kế trong việc phân tích, thiết kế xây dựng Hệ thống quản lý thẻ điện thoại trả trước tại Bưu điện Thành phố Hà Nội.

3.1 GIỚI THIỆU BÀI TOÁN

Trong sự phát triển chung, với đòi hỏi của thị trường ngày càng cao, các dịch vụ viễn thông được chú trọng hàng đầu để đem lại sự thuận lợi, thoải mái nhất cho người sử dụng. Các dịch vụ bán hàng trả trước ra đời đã đáp ứng được nhu cầu của một bộ phận lớn người tiêu dùng và đã có bước đột phát về số lượng khách hàng so với các dịch vụ khác cùng thời. Với sự tiện dụng khi hòa mạng ban đầu, với cách tính cước rõ ràng và sự chủ động cao khi sử dụng, các dịch vụ thẻ trả trước: Vinaphone card, Internet card chiếm ưu thế trong việc gia tăng số lượng người sử dụng hàng ngày các dịch vụ của Bưu điện Thành phố Hà Nội. Theo số liệu thống kê của Bưu điện Hà Nội, hiện nay số thuê bao di động trả trước chiếm tới gần 80% số thuê bao của mạng di động VinaPhone và MobilePhone.

Đối với các Bưu điện tỉnh thành như Bưu điện thành phố Hà Nội, việc nhận thẻ, bán thẻ và quản lý thẻ trước đây chủ yếu thực hiện trên sổ sách, số liệu được nhập và quản lý trên Hệ quản trị CSDL FoxPro một cách rời rạc. Tuy nhiên với sự phát triển nhanh của dịch vụ, việc giảm thiểu nhân sự và gia tăng không ngừng về số lượng thẻ, số lượng đại lý dẫn đến nhu cầu đối với một hệ thống quản lý đại lý và quản lý thẻ trả trước một cách thống nhất, tạo thuận lợi cho

đơn vị sử dụng là Phòng Kinh doanh - Công ty Viễn thông Hà Nội là nhu cầu cần thiết.

3.1.1 PHÁT BIỂU BÀI TOÁN

Tại Bưu điện Thành phố Hà Nội, hàng ngày Phòng kinh doanh căn cứ vào tình hình tiêu thụ để nhập loại thẻ trả trước từ các nhà cung cấp, sau đó xuất cho các đại lý kinh doanh thẻ - các đại lý này được Phòng Kinh doanh quản lý thông qua việc ký kết hợp đồng làm đại lý. Các đại lý kinh doanh thẻ đăng ký với phòng Kinh doanh về số lượng thẻ, chủng loại thẻ yêu cầu, ngoài ra có thể trao đổi thẻ tùy theo tình hình tiêu thụ. Phòng Kinh doanh sẽ phải luôn nắm được tình hình xuất, nhập thẻ của từng đại lý, tình hình tiêu thụ của mỗi loại thẻ để lập kế hoạch kinh doanh phù hợp. Ngoài ra Phòng Kinh doanh cũng cần theo dõi tình hình hoạt động của các đại lý như: nắm được các đại lý sắp hết hạn hợp đồng, đại lý kinh doanh không hiệu quả,... để có những chính sách phù hợp.

Bài toán đặt ra là cần xây dựng một hệ thống thực hiện quản lý việc nhập, xuất, phân phối, theo dõi kinh doanh thẻ trả trước tại Bưu điện thành phố Hà Nội.

3.1.2 CÁC THÀNH PHẦN CỦA HỆ THỐNG

Hệ thống quản lý thẻ gồm các thành phần: Chương trình quản lý thẻ và theo dõi kinh doanh, chương trình dịch vụ quản lý yêu cầu và số liệu báo cáo từ đại lý, trình nhập số liệu từ đại lý. Trong đó:

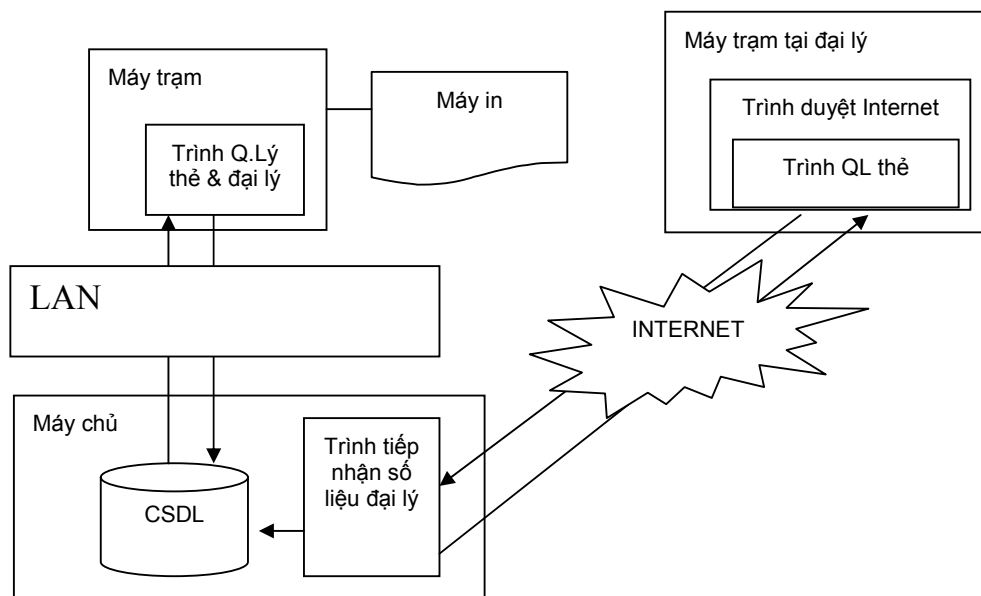
- Chương trình quản lý thẻ và theo dõi kinh doanh cho phép nhập số liệu về thẻ và đại lý, theo dõi và quản lý các đại lý,... Chương trình được cài đặt trên máy trạm trong hệ thống mạng nội bộ của Bưu điện.

- Chương trình dịch vụ quản lý yêu cầu và số liệu báo cáo từ đại lý; Thực hiện việc tiếp nhận yêu cầu hoặc số liệu từ đại lý và thông báo thông tin của Bưu điện cho các đại lý. Chương trình được cài đặt trên máy chủ của Bưu điện.

- Chương trình nhập số liệu từ đại lý chạy trên máy cá nhân của đại lý có kết nối Internet với máy chủ của Bưu điện. Chương trình cho phép các đại lý đăng ký mua thẻ và báo cáo số liệu kinh doanh, thông báo cho đại lý các thông tin/yêu cầu của Bưu điện đối với từng đại lý.

3.1.3 KIẾN TRÚC MÔI TRƯỜNG HỆ THỐNG

Hệ thống quản lý thẻ trả trước (hay Hệ thống quản lý thẻ) được yêu cầu thiết kế với các chức năng chính về quản lý thẻ và đại lý chạy trên môi trường mạng cục bộ tại Bưu điện Hà Nội. Với đại lý bán thẻ phải có khả năng cung cấp các số liệu bán hàng của đại lý, gửi yêu cầu đặt mua thẻ và nhận các yêu cầu của Bưu điện qua mạng Internet. Hình 3.1 thể hiện kiến trúc môi trường Hệ thống quản lý thẻ của Bưu điện Hà Nội. Luận văn tập trung phân tích thiết kế Chương trình quản lý thẻ và đại lý chạy trên môi trường mạng cục bộ tại Bưu điện.



Hình 3.1 Kiến trúc môi trường Hệ thống quản lý thẻ của Bưu điện Hà Nội

3.2 THU THẬP VÀ PHÂN TÍCH YÊU CẦU – MÔ HÌNH USE CASE

3.2.1 MỤC TIÊU CỦA HỆ THỐNG

Hệ thống Quản lý thẻ có mục tiêu phục vụ việc quản lý một cách đồng bộ các hoạt động kinh doanh thẻ cũng như quản lý mạng lưới đại lý bán thẻ trên toàn địa bàn, đảm bảo an toàn thông tin, đáp ứng kịp thời và chính xác các yêu cầu về báo cáo số liệu để phục vụ việc quản lý và hoạch định chính sách kinh doanh của Bưu điện Thành phố Hà Nội.

3.2.2 ĐẶC TẢ CÁC CHỨC NĂNG HỆ THỐNG

Các chức năng của hệ thống được chia làm hai loại:

- Chức năng rõ: Là chức năng mà người sử dụng có nhìn thấy trên giao diện của hệ thống và có thể thực hiện.
- Chức năng ẩn: Thường là chức năng kỹ thuật, người sử dụng ít nhận thấy và không được thể hiện trên giao diện của hệ thống.

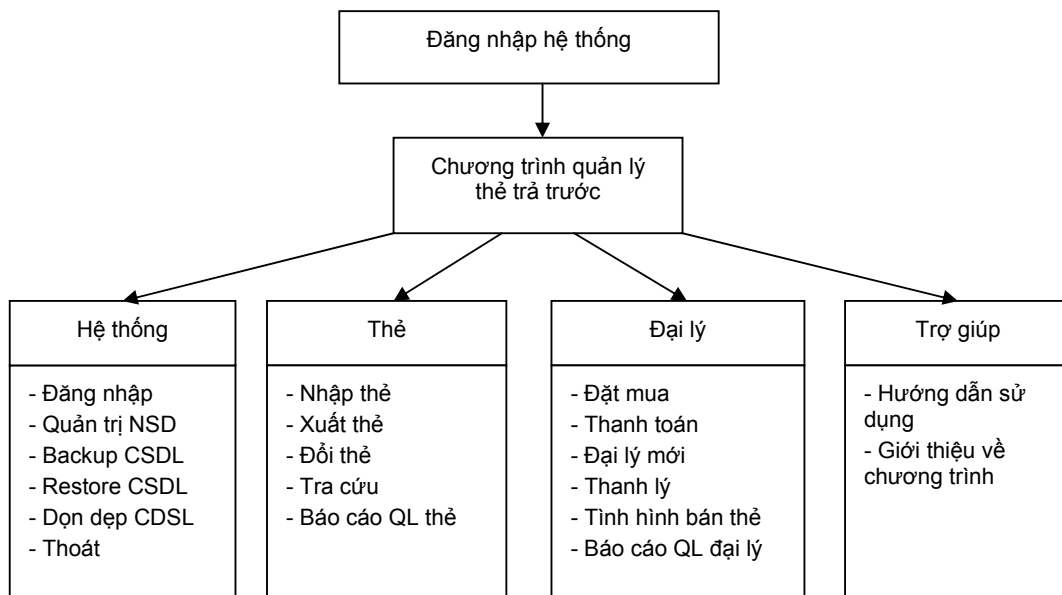
Các chức năng cơ bản của Hệ thống quản lý thẻ được thể hiện trong bảng 3.1.

Mã	Chức năng	Loại
R1	Quản lý hệ thống	
R1.1	Đăng nhập: Quản lý việc đăng nhập hệ thống, nhập user và password, hệ thống kiểm tra và nhận dạng, nếu thành công sẽ tự động kết nối	rõ
R1.2	Quản trị người sử dụng: Cho phép nhập mới, điều chỉnh, xóa thông tin về người sử dụng, các quyền truy nhập vào từng thành phần của hệ thống (thẻ , đại lý, quyền sửa đổi, chỉ xem, quản trị)	rõ
R1.3	Backup CSDL: Định kỳ sao lưu các dữ liệu để đảm bảo an toàn cho hệ thống, đây là chức năng ẩn, dành cho người quản trị hệ thống	ẩn

Mã	Chức năng	Loại
R1.4	Khôi phục CSDL: Khôi phục dữ liệu khi có sự cố	rõ
R1.5	Dọn dẹp CSDL: Định kỳ dọn dẹp, tránh những số liệu “rác” để hệ thống hoạt động một cách hiệu quả, Chức năng này được tự động thực hiện định kỳ hoặc do người quản trị hệ thống thực hiện	ẩn/rõ
R2	Quản lý thẻ	
R2.1	Nhập thẻ: Nhập thẻ từ nhà cung cấp. Nhập thông tin khi nhận thẻ từ nhà cung cấp, xem thông tin về các đợt nhập thẻ. Cho phép điều chỉnh các thông tin nhập sai.	rõ
R2.2	Xuất thẻ: Xuất thẻ cho các đại lý, xem thông tin về các đợt xuất thẻ. Cho phép điều chỉnh các thông tin sai.	rõ
R2.3	Đổi thẻ: Cập nhập thông tin về việc đổi lại thẻ đối với các đại lý theo nhu cầu thường xuyên.	rõ
R2.4	Tra cứu: Tra cứu thông tin xuất thẻ khi nhập số seri tương ứng.	rõ
R2.5	Báo cáo (BC) quản lý thẻ: BC nhập hàng tổng hợp, BC nhập hàng chi tiết, BC xuất hàng tổng hợp cho từng đại lý, BC xuất hàng chi tiết cho từng đại lý, BC xuất hàng tổng hợp tất cả các đại lý, BC tình hình kinh doanh	rõ
R3	Quản lý đại lý bán thẻ	
R3.1	Đặt mua: Đại lý đặt mua thẻ qua mạng hoặc qua điện thoại.	
R3.2	Thanh toán: Đại lý thanh toán tiền mua thẻ cho Bưu điện.	
R3.3	Đại lý mới: Đăng ký đại lý mới và nhập các thông tin về đại lý mới, cho phép điều chỉnh thông tin.	rõ
R3.4	Thanh lý: Nhập thông tin về các đại lý hết hạn hợp đồng cần thanh lý	rõ
R3.5	Tình hình bán thẻ: Nhập thông tin về tình hình bán thẻ của đại lý	rõ
R3.6	Báo cáo về quản lý đại lý: Danh sách các đại lý đã hết hạn hợp đồng, danh sách các đại lý còn hạn hợp đồng	rõ

Bảng 3.1 Các chức năng cơ bản của Hệ thống quản lý thẻ

Sơ đồ khối chức năng hệ thống được thể hiện như trong hình 3.2.



Hình 3.2 Sơ đồ chức năng của Hệ thống

3.2.3 NHẬN BIẾT VÀ MÔ TẢ CÁC TÁC NHÂN VÀ TRƯỜNG HỢP SỬ DỤNG

3.2.3.1 Các tác nhân - Actors

Các tác nhân của hệ thống quản lý thẻ bao gồm: Giám đốc, người quản trị hệ thống, nhân viên quản lý, nhân viên kế toán và chủ đại lý bán thẻ.

1. *Giám đốc (Giamdoc)*: Là người theo dõi điều hành hoạt động kinh doanh, có quyền truy cập các chức năng báo cáo của hệ thống.

2. *Người quản trị hệ thống (QuantriHT)*: Là nhân viên, được giao quyền giám sát hoạt động của hệ thống, cấp phát quyền truy nhập, cập nhật thông tin về hệ thống.

3. *Nhân viên quản lý (NhanvienQL)*: Là nhân viên, được cấp quyền truy nhập hệ thống, tùy theo trách nhiệm có thể được nhập dữ liệu, xem hay điều chỉnh các dữ liệu thẻ.

4. *Nhân viên kế toán (NhanvienKT)*: Là kế toán viên, được sử dụng chức năng thanh toán thực hiện nhiệm vụ thanh toán tiền mua thẻ với đại lý bán thẻ.

5. *Chủ đại lý (ChuDaily)*: Là chủ các đại lý bán thẻ, được cấp quyền truy cập hệ thống từ xa để nhập yêu cầu mua thẻ và báo cáo kết quả bán thẻ.

3.2.3.2 Các trường hợp sử dụng - Use cases

Trường hợp sử dụng (*User case - UC*) mô tả ‘ai’ sử dụng hệ thống như thế nào, mô tả tương tác giữa người sử dụng với hệ thống phần mềm để thực hiện các thao tác giải quyết một công việc cụ thể nào đó.

Các UC của hệ thống được xác định dựa vào các tác nhân hay dựa vào các sự kiện. Với mỗi tác nhân tìm những tiến trình khởi đầu, các tiến trình giúp tác nhân giao tiếp, tương tác với hệ thống. Đối với sự kiện, xác định sự kiện bên ngoài có tác động đến hệ thống hay hệ thống phải trả lời, tìm mối liên quan giữa các sự kiện và các tác nhân.

Trên cơ sở các thao tác của tác nhân, các sự kiện và chức năng cơ bản, Hệ thống quản lý thẻ tại Bưu điện Hà Nội có ba nhóm các UC cơ bản: các UC phục vụ việc quản lý hệ thống, các UC phục vụ việc quản lý thẻ, và các UC phục vụ việc quản lý đại lý bán thẻ tương ứng với 3 chức năng tổng quát của hệ thống.

R1. Các UC phục vụ việc quản lý hệ thống

R1.1) Đăng nhập

Tên: Dangnhap

Tác nhân: Giám đốc, người quản trị hệ thống, nhân viên quản lý, nhân viên thu ngân.

Mục đích: Kiểm tra, nhận dạng người đăng nhập vào hệ thống.

Mô tả khái quát: Người sử dụng (tác nhân) đăng nhập vào hệ thống, hệ thống nhận dạng và cho phép thực hiện các chức năng theo quyền được cấp của tác nhân.

R1.2) Quản trị người sử dụng

Tên: QuantriNSD

Tác nhân: Người quản trị hệ thống

Mục đích: Cấp phát ID, mật khẩu và quyền truy nhập truy nhập vào từng chức năng của hệ thống cho từng người sử dụng; Cho phép kiểm tra nhập ký sử dụng hệ thống của từng người sử dụng.

Mô tả khái quát: Người quản trị hệ thống gọi thực hiện chức năng. Cấp phát, thay đổi hay loại bỏ ID, mật khẩu, quyền của từng người sử dụng. Kiểm tra nhập ký truy nhập hệ thống.

R1.3) Backup CSDL

Tên: BackupDL

Tác nhân: Người quản trị hệ thống.

Mục đích: Sao lưu lại các số liệu của hệ thống để đảm bảo không bị mất số liệu hoặc số liệu bị mất là ít nhất nếu hệ thống bị rủi ro.

Mô tả khái quát: Hệ thống tạo ra một bản sao của CSDL trên máy chủ sau mỗi giao dịch hay do người quản trị hệ thống ra lệnh thực hiện.

R1.4) Khôi phục CSDL

Tên: KhoiphucDL

Tác nhân: Người quản trị hệ thống.

Mục đích: Khôi phục lại số liệu gần nhất nếu có sự cố về CSDL để đảm bảo hệ thống hoạt động bình thường.

Mô tả khái quát: Chức năng sẽ tạo lại các số liệu cho hệ thống từ các số liệu được được sao lưu trước đó.

R1.5) Dọn dẹp CSDL:

Tên: DondepDL

Tác nhân: Người quản trị hệ thống.

Mục đích: Dọn dẹp dữ liệu trong CSDL, loại bỏ những số liệu “rác” để hệ thống hoạt động một cách hiệu quả.

Mô tả khái quát: Kiểm tra phát hiện và loại bỏ các bản ghi tạm thời nảy sinh trong quá trình xử lý số liệu hay những bản ghi được đánh dấu xóa bỏ. Chức năng này được tự động thực hiện định kỳ hoặc do người quản trị hệ thống thực hiện.

R2. Các Use case phục vụ quản lý thẻ

R2.1) Nhập thẻ

Tên: NhapThe

Tác nhân: Nhân viên quản lý

Mục đích: Nhập (mua) thẻ từ nhà cung cấp.

Mô tả khái quát: Nhân viên quản lý gọi thực hiện chức năng này để nhập thẻ từ nhà cung cấp.

R2.2) Xuất thẻ

Tên: XuatThe

Tác nhân: Nhân viên quản lý, Chủ đại lý

Mục đích: Xuất (bán) thẻ cho đại lý.

Mô tả khái quát: Nhân viên quản lý gọi thực hiện chức năng này để xuất thẻ cho đại lý.

R2.3) Đổi thẻ

Tên: DoiThe

Tác nhân: Nhân viên quản lý

Mục đích: Đổi lại thẻ đã xuất đối với các đại lý theo nhu cầu thường xuyên.

R2.4) Tra cứu thẻ

Tên: TracuuThe

Tác nhân: Nhân viên quản lý, quản trị hệ thống, giám đốc

Mục đích: Tra cứu thông tin xuất/nhập thẻ khi nhập số seri tương ứng.

R2.5 Báo cáo quản lý thẻ

Tên: BaocaoQLThe

Tác nhân: Nhân viên quản lý, giám đốc

Mục đích: Lập các báo cáo về việc quản lý thẻ: BC nhập hàng tổng hợp, BC nhập hàng chi tiết, BC xuất hàng tổng hợp cho từng đại lý, BC xuất hàng chi tiết cho từng đại lý, BC xuất hàng tổng hợp tất cả các đại lý, BC tình hình kinh doanh.

R3. Các Use case quản lý đại lý

R3.1 Đặt mua thẻ

Tên: DatmuaThe

Tác nhân: Nhân viên quản lý, Chủ đại lý

Mục đích: Nhập yêu cầu đặt mua thẻ của đại lý

Mô tả khái quát: Nhân viên quản lý gọi thực hiện chức năng này để nhập yêu cầu đặt mua của đại lý (ở đây yêu cầu của đại lý được thông báo qua điện thoại).

R3.2) Thanh toán

Tên: Thanhtoan

Tác nhân: Nhân viên kế toán

Mục đích: Đại lý trả tiền mua thẻ cho bưu điện.

Mô tả khái quát: Nhân viên kế toán thu tiền mua thẻ của đại lý. Việc thanh toán có thể bằng tiền mặt, thẻ tín dụng hoặc séc. Đại lý được phép có thể không phải trả hết số tiền thẻ đã mua (được phép nợ lại một phần).

R3.3) Đại lý mới

Tên: Dailymoi

Tác nhân: Nhân viên quản lý

Mục đích: Đăng ký đại lý mới

Mô tả khái quát: Nhập thông tin về các đại lý mới ký hợp đồng bán thẻ.

R3.4) Thanh lý

Tên: Thanhly

Tác nhân: Nhân viên quản lý

Mục đích: Thanh lý hợp đồng làm đại lý.

Mô tả khái quát: Nhập thông tin về các đại lý hết hạn hợp đồng làm đại lý bán thẻ.

R3.5) Tình hình bán thẻ

Tên: TinhhhinhBanthe

Tác nhân: Nhân viên quản lý, giám đốc, chủ đại lý

Mục đích: Theo dõi việc bán thẻ của đại lý

Mô tả khái quát: Theo dõi và cập nhật thông tin về tình hình bán thẻ của đại lý.

R3.6) Báo cáo quản lý đại lý

Tên: BaocaoQLDaily

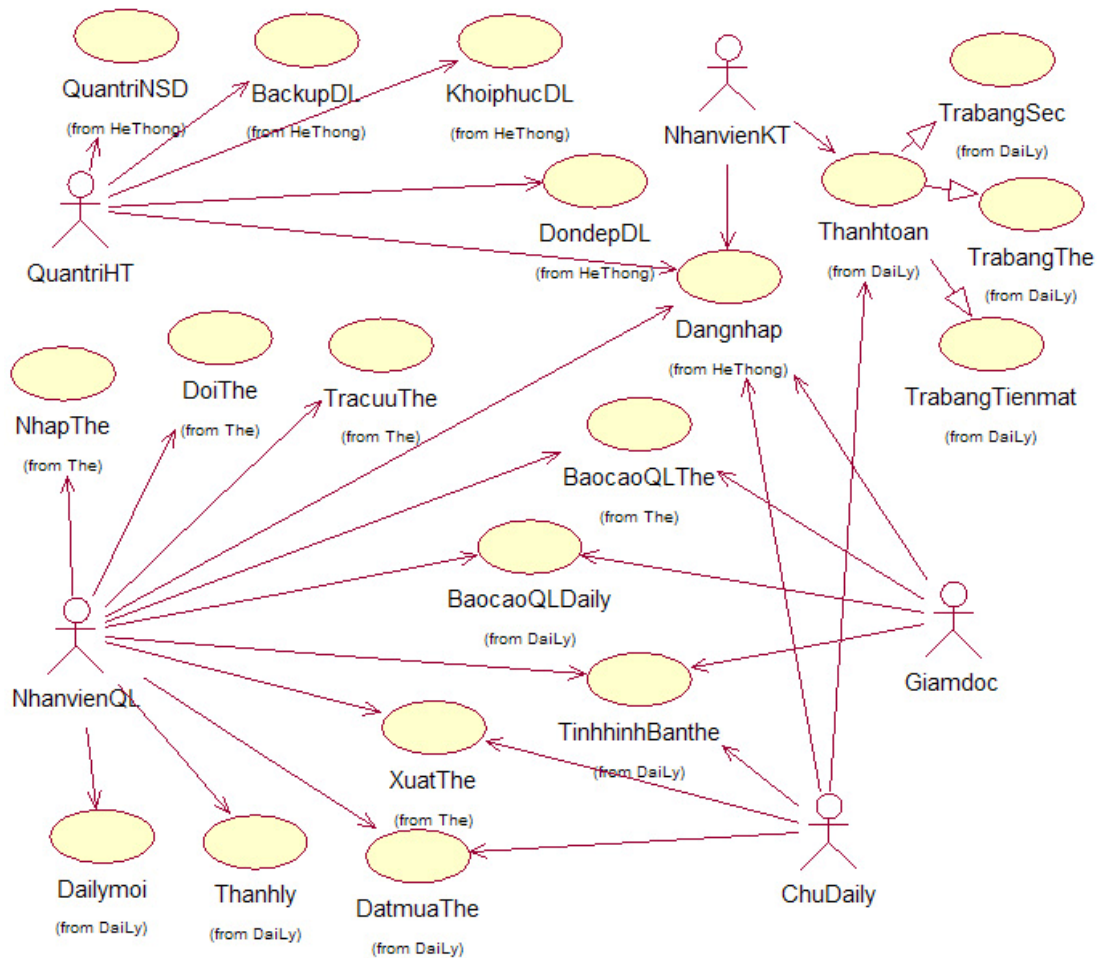
Tác nhân: Nhân viên quản lý, giám đốc

Mục đích: Lập báo cáo về tình hình quản lý đại lý

Mô tả khái quát: Báo cáo danh sách các đại lý đã hết hạn hợp đồng, danh sách các đại lý còn hạn hợp đồng.

3.2.4 BIỂU ĐỒ USE CASES

Hình 3.3 là biểu đồ UC của hệ thống quản lý thẻ



Hình 3.3 Biểu đồ UC của hệ thống

3.2.5 MÔ HÌNH HÓA NGHIỆP VỤ

Trong khuôn khổ của luận văn, học viên tập trung ứng dụng phương pháp hướng đối tượng và mẫu để thiết kế ba chức năng chính: **nhập thẻ**, **xuất thẻ** và **thanh toán**.

i) Chi tiết hóa diễn biến sự kiện của UC Nhập thẻ

Tên: NhapThe

Tác nhân: Nhân viên quản lý

Ghi chú: Hiện tại chỉ có một nhà cung cấp thẻ duy nhất nên việc nhập thẻ được hiểu là nhập từ nhà cung cấp này. Việc nhập thẻ được thực hiện thông qua hộp thoại nhập thẻ như hình 3.4.

Hình 3.4 Giao diện cho việc nhập thẻ

Diễn biến sự kiện:

<i>Hành động của tác nhân</i>	<i>Phản hồi của hệ thống</i>
1. Thực hiện chức năng	2. Hiện thị hộp thoại giao diện nhập thẻ
3. Chọn loại thẻ, nhập số lượng (từ seri1 đến seri2) và các thông tin liên quan. Khi nhập xong dữ liệu chọn chức năng nhập (ví dụ bấm phím Nhap) để kết thúc việc nhập cho một loại thẻ.	
4. Lặp lại bước 3 để nhập cho loại thẻ tiếp theo.	
5. Kết thúc việc nhập thẻ bằng việc chọn chức năng kết thúc (ví dụ: bấm phím kết thúc).	6. Hiện thị hộp thoại xác nhận kết thúc việc nhập thẻ tốt đẹp.
7. Xác nhận có/không (hủy bỏ).	8. Nếu không đồng ý chuyển đến bước 13 (kết thúc).

	9. Tạo phiếu nhập, thực hiện yêu cầu, cập nhật CSDL. 10. Backup dữ liệu
	11. Ghi nhật ký làm việc
	12. Thông báo kết quả thực hiện
13. Kết thúc công việc	

ii) Chi tiết hóa diễn biến sự kiện của UC Xuất thẻ

Tên: XuatThe

Tác nhân: Nhân viên quản lý, Chủ đại lý

Ghi chú: Việc xuất thẻ được thực hiện thông qua hộp thoại xuất thẻ như hình 3.5.

Hình 3.5 Giao diện cho việc xuất thẻ

Diễn biến sự kiện:

<i>Hành động của tác nhân</i>	<i>Phản hồi của hệ thống</i>
1. Thực hiện chức năng	2. Hiện thị hộp thoại giao diện xuất thẻ
3.1 Chọn đại lý; 3.2 Chọn loại thẻ; nhập số lượng xuất (từ seri1 đến seri2) và các thông tin liên quan. Khi nhập xong chọn chức năng xuất để ghi nhận việc xuất loại thẻ được chọn	4. Kiểm tra lượng thẻ xuất > lượng thẻ có và số seri có còn trong kho không? Nếu lượng thẻ xuất > lượng thẻ có hoặc số seri không còn trong kho: thông báo lỗi

5. Nhập lại số lượng xuất nếu có thông báo lỗi về số lượng	
6. Lặp lại bước 3.2 để xuất loại thẻ tiếp theo	
7. Kết thúc việc xuất thẻ bằng việc chọn chức năng kết thúc (ví dụ: bấm phím kết thúc)	8. Hiện thị hộp thoại xác nhận việc xuất thẻ
9. Xác nhận đồng ý/không đồng ý	10. Nếu không đồng ý chuyển đến bước cuối cùng. 12. Tạo phiếu xuất. 13. Tính tổng tiền của hóa đơn
14. Đưa thẻ cho đại lý	15. Cập nhật CSDL 16. Tạo bản backup dữ liệu 17. Ghi nhật ký làm việc 18. Thông báo kết quả thực hiện
18. Kết thúc công việc	

iii) Chi tiết hóa diễn biến sự kiện của UC Thanh toán

Tên: Thanhtoan

Tác nhân: Nhân viên kế toán, Chủ đại lý

Ghi chú: Việc thanh toán được thực hiện thông qua hộp thoại thanh toán như hình 3.6.

Hình 3.6 Giao diện cho việc thanh toán

Diễn biến sự kiện:

<i>Hành động của tác nhân</i>	<i>Phản hồi của hệ thống</i>
1. Kế toán thực hiện chức năng	2. Hiện thị hợp thoại giao diện thanh toán
3. Kế toán chọn đại lý	4. Tính tổng tiền đại lý còn nợ 5. Hiện thị tổng tiền nợ
6. Kế toán thông báo cho chủ đại lý số tiền còn nợ	
7. Chủ đại lý Chọn phương thức thanh toán i) Nếu trả bằng tiền mặt: xem diễn biến sự kiện thu bằng tiền mặt ii) Nếu trả bằng thẻ tín dụng: xem diễn biến sự kiện thu bằng thẻ tín dụng iii) Nếu trả bằng Séc: xem diễn biến sự kiện thu bằng Séc	8. Hiện thị số dư hoặc số tiền còn thiếu của đại lý 9. Cập nhật thông tin thanh toán 10. Tạo bản sao dữ liệu 11. Ghi nhật ký làm việc 12. Thông báo kết quả thực hiện
13. Kết thúc phiên thanh toán	

iv) Trường hợp Thanh toán bằng tiền mặt:

Tên: TrabangTienmat(tientra: Number)

Tác nhân: Nhân viên kế toán, Chủ đại lý

Diễn biến sự kiện:

<i>Hành động của tác nhân</i>	<i>Phản hồi của hệ thống</i>
1 Trường hợp sử dụng này được bắt đầu khi Chủ đại lý chọn hình thức thanh toán bằng tiền mặt sau khi được thông báo số tiền đại lý còn nợ	
1. Chủ đại lý trả đưa tiền mặt cho nhân viên kế toán	
3. Kế toán nhận tiền và nhập vào số tiền đại lý trả	4. Hiện thị số dư hoặc số tiền còn thiếu của đại lý (trường hợp còn

	thiếu, đại lý vẫn nợ lại một phần tiền mua thẻ)
5. Kế toán trả lại số tiền còn dư hoặc thông báo cho chủ đại lý số tiền còn nợ	

v) *Trường hợp Thanh toán bằng thẻ tín dụng:*

Tên: TrabangThe

Tác nhân: Nhân viên kế toán, Chủ đại lý

Diễn biến sự kiện:

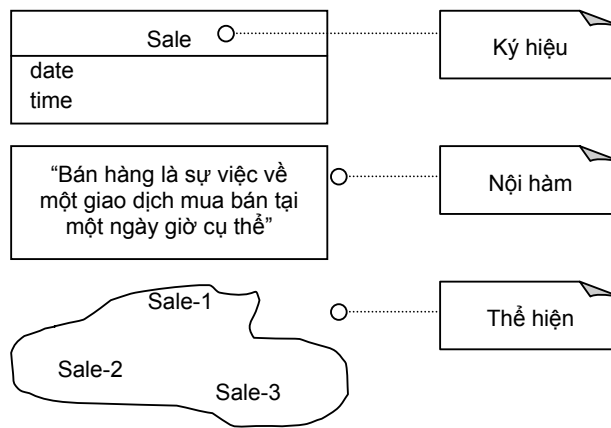
<i>Hành động của tác nhân</i>	<i>Phản hồi của hệ thống</i>
1. Chủ đại lý trả bằng thẻ tín dụng	2. Phát sinh yêu cầu gửi tới bộ phận kiểm tra thẻ tín dụng.
3. Bộ phận kiểm tra thẻ cho phép trả tiền bằng thẻ tín dụng sau khi kiểm tra.	4. Trừ số tiền phải trả vào tài khoản tín dụng

Diễn biến sự kiện thanh toán bằng Séc tương tự như việc thanh toán bằng thẻ tín dụng.

3.3 THU THẬP VÀ PHÂN TÍCH YÊU CẦU - MÔ HÌNH KHÁI NIỆM

Khái niệm được hiểu là một ý tưởng, đồ vật hay một đối tượng. Một khái niệm có thể được mô tả bởi ký hiệu, nội hàm và ngoại diên của nó. Ví dụ: Khái niệm bán hàng (sale) được mô tả như hình 3.7.

Mô hình khái niệm là sự trình bày các khái niệm trong lĩnh vực vấn đề nào đó. Trong ngôn ngữ UML, mô hình khái niệm là minh họa với một tập các biểu đồ cấu trúc tĩnh mà trong đó không mô tả các thao tác.



Hình 3.7: Khái niệm bao gồm ký hiệu, nội hàm và thể hiện

3.3.1 NHẬN BIẾT CÁC KHÁI NIỆM (ĐỐI TƯỢNG)

Các khái niệm của hệ thống có thể được nhận biết theo hai cách cơ bản như sau:

(i) Xác định các danh từ trong mô tả văn bản của bài toán, các danh từ này có thể là đại biểu của lớp hay thuộc tính của lớp. Sau đó dựa vào kiến thức thực tế bỏ đi những mục không phải là đại biểu của lớp.

(ii) Dựa vào mục đích của các trường hợp sử dụng để xác định các lớp đối tượng.

- Xác định mục đích của UC bằng cách xác định mục tiêu, dịch vụ, những giá trị hay những đáp ứng mà UC đó có thể cung cấp. Tiếp theo xác định các thực thể (class) tham gia thực hiện các mục tiêu của UC bằng cách xét những thực thể và những thuộc tính quan trọng và cần thiết để thực hiện các UC, từ đó đặt tên và gán trách nhiệm cho lớp đề cử (mỗi khái niệm lập một lớp và đặt tên cho nó).

- Xác định các mối quan hệ giữa các lớp: Mỗi quan hệ là các phát hiện khởi đầu cho các liên kết và thuộc tính.

- Xác định các thành phần thể hiện sự cộng tác của các lớp trong UC.

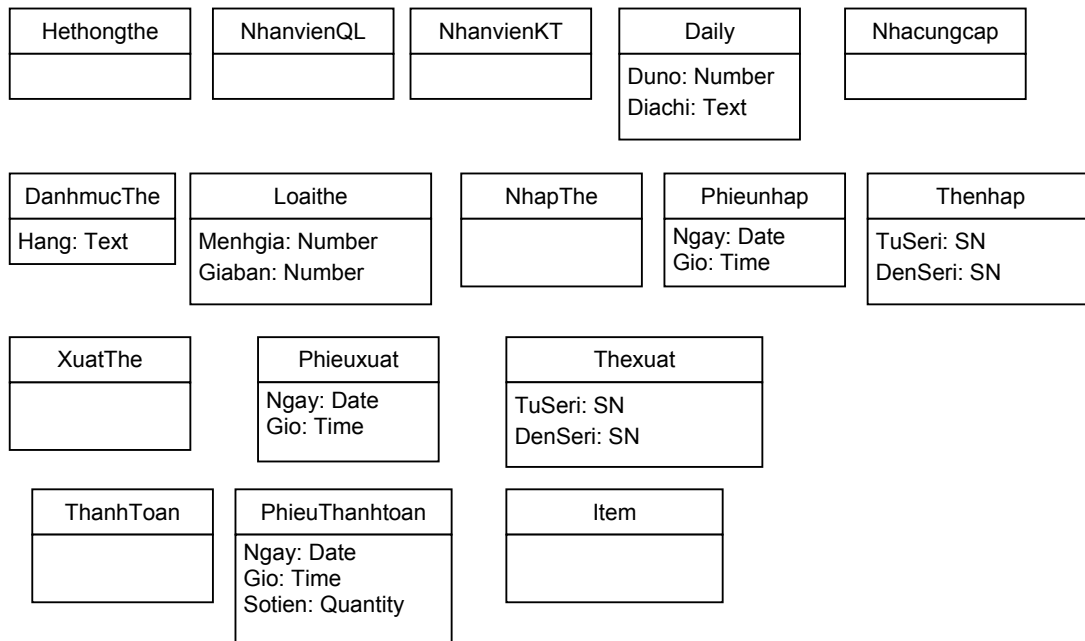
- Kiểm tra các biểu đồ được xây dựng: Kiểm tra các yêu cầu chức năng xem các UC thực hiện hết yêu cầu chưa? mục đích của UC có đúng không? Và kiểm tra các thực thể trong biểu đồ lớp có cần và đủ để thực hiện mục đích của UC không? Các thuộc tính của thực thể có phải cái mà UC cần biết không? Các hàm thành phần của thực thể có cần và đủ để thực hiện mục đích của mỗi UC không?

Bằng hai cách trên xác định được hệ thống quản lý thẻ điện thoại với các chức năng chính như trên có các khái niệm sau:

Hethongthe (hệ thống quản lý)	Phieunhap (phiếu nhập)
NhanvienQL (nhân viên quản lý)	Thenhap (danh sách loại thẻ được nhập)
NhanvienKT (nhân viên kế toán)	XuatThe (phiên xuất thẻ)
Daily (đại lý bán thẻ)	Phieuxuat (phiếu xuất)
Nhacungcap (công ty cung cấp các loại thẻ)	Thexuat (danh sách loại thẻ được xuất)
DanhmucThe (danh mục các loại thẻ: Vina, Mobile, Viettel,...)	Thanhtoan (phiên thanh toán)
Loaithe (loại thẻ: 500K, 200K, ...)	Phieuthanhtoan (hoán đơn thanh toán)
NhapThe (phiên nhập thẻ)	Item (thẻ)

3.3.2 THUỘC TÍNH CỦA CÁC LỚP

Từ thực tế, ta có thể xác định được thuộc tính cho các lớp trong hệ thống như trong hình 3.8.



Hình 3.8 Thuộc tính cơ bản của các lớp

3.3.3 NHẬN BIẾT CÁC QUAN HỆ CÁC KHÁI NIỆM

Trong hệ thống ta có quan hệ giữa các lớp như sau:

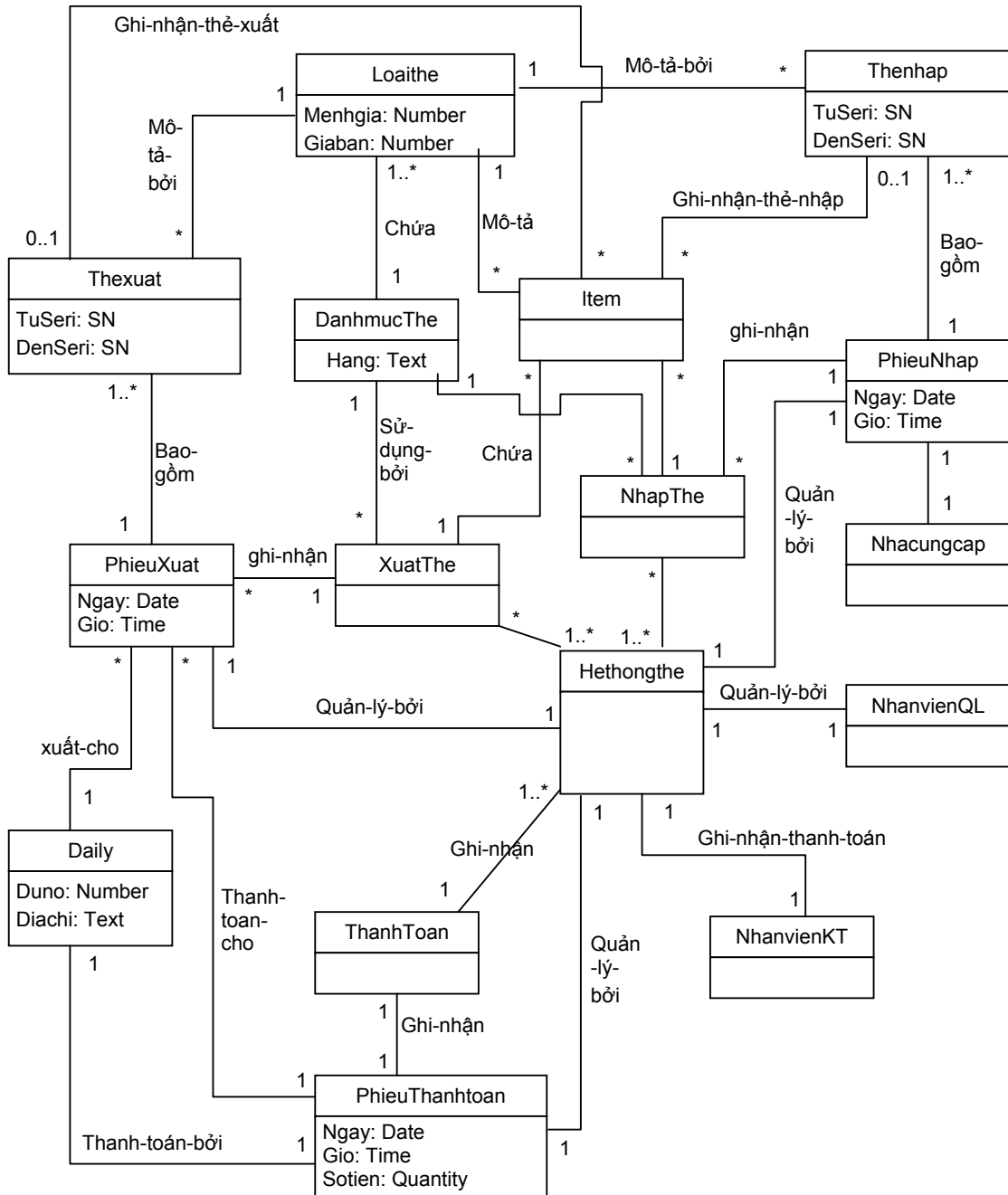
NhanvienQL - Hethongthe	ThanhToan - Phieuthanhtoan
NhanvienKT - Hethongthe	Phieuthanhtoan - Daily
Hethongthe - XuatThe	Hethongthe - NhapThe
Hethongthe - Phieuxuat	Hethongthe - Phieunhap
XuatThe - Danh mucThe	NhapThe - Danh mucThe
XuatThe - Phieuxuat	NhapThe - Phieunhap
XuatThe - Item	NhapThe - Item
Phieuxuat - Thexuat	Phieunhap - Thexuat
Phieuxuat - Daily	Phieunhap - Nhacungcap
Phieuxuat - ThanhToan	Thenhap - Loaithe
Thexuat - Loaithe	Thenhap - Item
Thexuat - Item	Loaithe - Danh mucThe

Hethongthe - ThanhToan

Item - Loaithe

Hethongthe - Phieuthanhtoan

Tổng hợp các quan hệ giữa các khái niệm như phân tích ở trên ta có biểu đồ quan hệ giữa các khái niệm như hình 3.9



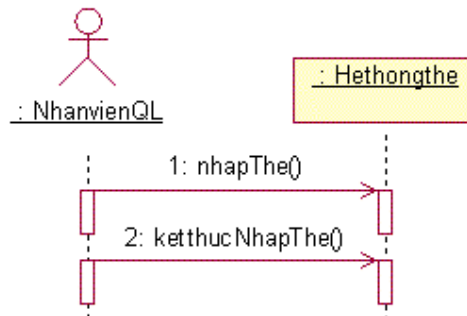
Hình 3.9 Biểu đồ quan hệ giữa các khái niệm

3.4 HÀNH VI HỆ THỐNG - CÁC BIỂU ĐỒ TRÌNH TỰ

3.4.1 BIỂU ĐỒ TRÌNH TỰ HỆ THỐNG

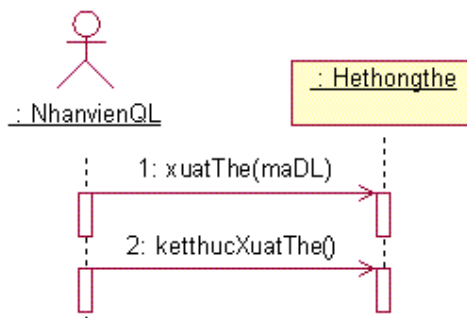
i) Biểu đồ trình tự việc nhập thẻ

Biểu đồ trình tự nhập thẻ được thể hiện trong hình 3.10. Khi nhập xong các thông tin cho việc nhập một loại thẻ, đối tượng :nhanvienQL gửi đến :Hethongthe thông điệp `nhapThe()` (chọn phím ‘Nhập’). Kết thúc việc nhập thẻ bằng cách nhấn phím ‘ket thuc’ (gửi cho hệ thống thông điệp `ketthucNhapThe()`). Hệ thống thực hiện cập nhật dữ liệu.



Hình 3.10 Biểu đồ trình tự nhập thẻ

ii) Biểu đồ trình tự xuất thẻ



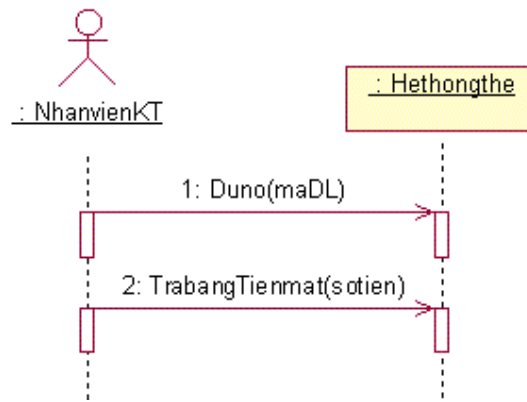
Hình 3.11 Biểu đồ trình tự xuất thẻ

Biểu đồ trình tự xuất thẻ được thể hiện trong hình 3.11. Khi nhập xong các thông tin cho một loại thẻ xuất, đối tượng :nhanvienQL gửi đến :Hethongthe

thông điệp `xuatThe(maDL)` với mã của đại lý (`maDL`). Kết thúc việc nhập thẻ bằng cách gửi cho hệ thống thông điệp `ketthucXuatThe()` (nhấn phím ‘`ket thuc`’). Hệ thống thực hiện cập nhật dữ liệu.

iii) Biểu đồ trình tự thanh toán bằng tiền mặt

Hình 3.12 thể hiện biểu đồ trình tự mô tả hoạt động của nhân viên kế toán và hệ thống thực hiện việc thanh toán bằng tiền mặt.



Hình 3.12 Biểu đồ trình tự thanh toán bằng tiền mặt

Sau khi nhập thông tin về đại lý, nhân viên kế toán gửi đến :Hethongthe thông điệp `Duno(maDL)` với mã đại lý (`maDL`) để kiểm tra số tiền đại lý còn nợ. Hệ thống tính và thông báo tổng số tiền mà đại lý phải trả (tổng số tiền đại lý còn nợ). Nếu đại lý trả bằng tiền mặt thì :nhanvienKT gửi tiếp đến cho :Hethongthe thông điệp `TrabangTienmat(num)`.

Nếu đại lý trả bằng Séc hoặc thẻ tín dụng thì gửi đến cho :Hethongthe thông điệp `TrabangSec(driverLicenseNum)` hay `TrabangThe(ccNum, expirryDate)`.

3.4.2 GIAO KÈO THAO TÁC CỦA HỆ THỐNG

Trong biểu đồ trình tự đã cho biết tên của những nhiệm vụ mà mỗi đối tượng phải thực hiện khi nhận được thông điệp yêu cầu, tuy nhiên nó chưa chỉ rõ

cách thức thực hiện việc đó như thế nào. Việc xây dựng các giao kèo thao tác của hệ thống, còn gọi là hợp đồng tương tác hay các đặc tả cho những hoạt động (thủ tục, hàm) sẽ chỉ rõ các hành vi của các đối tượng (mô tả cái gì sẽ xảy ra) và hỗ trợ cho việc thiết kế và cài đặt các lớp.

3.4.2.1 Hợp đồng nhập thẻ

<i>Tên gọi:</i>	<code>nhapThe(maloai:text, tuSer:i:SN, denSer:i:SN)</code>
<i>Trách nhiệm:</i>	Ghi thông tin về lô thẻ nhập (bao gồm loại thẻ, số sê-ri bắt đầu, số sê-ri kết thúc), bổ sung nó vào phiếu nhập.
<i>Kiểu / Lớp:</i>	System (Hệ thống)
<i>Tham chiếu tới:</i>	Use case nhập thẻ
<i>Chú ý:</i>	Sử dụng phương pháp truy nhập nhanh vào CSDL
<i>Ngoại lệ:</i>	Nếu số sê-ri bắt đầu (<code>tuSer:i</code>) lớn hơn số sê-ri kết thúc (<code>denSer:i</code>) thì báo lỗi
<i>Post-conditions:</i>	+ Nếu là phiên nhập mới thì tạo ra một đối tượng <code>Phieunhap</code> và kết hợp nó vào <code>Hethongthe</code> + Tạo ra một đối tượng <code>Thexuat</code> cho loại thẻ vừa nhập vào và liên kết nó với <code>Phieunhap</code>. + Các giá trị nhập vào được gán cho các thuộc tính của đối tượng <code>Thenhap</code> (<code>tuSer:i</code>, <code>denSer:i</code>,...) + <code>Thenhap</code> được liên kết với <code>Loaithe</code> dựa vào mã của loại thẻ nhập (<code>maloai</code>)

3.4.2.2 Hợp đồng kết thúc phiên nhập thẻ

<i>Tên gọi:</i>	<code>ketthucNhapthe()</code>
<i>Trách nhiệm:</i>	Kết thúc phiên nhập thẻ, ghi dữ liệu vào CSDL.
<i>Kiểu / Lớp:</i>	System (Hệ thống)
<i>Tham chiếu tới:</i>	Use case nhập thẻ
<i>Ngoại lệ:</i>	Nếu chưa có loại thẻ nào được nhập thì báo lỗi

Post-conditions: Chuyển trạng thái của phiên nhập thẻ sang kết thúc. Cập nhật CSDL

3.4.2.3 Hợp đồng xuất thẻ

Tên gọi: `xuatThe(maDL: text, maloai:text, tuSeri:SN, denSeri:SN)`

Trách nhiệm: Ghi thông tin về lô thẻ xuất (bao gồm loại thẻ, số sê-ri bắt đầu, số sê-ri kết thúc), bổ sung nó vào phiếu xuất.

Kiểu / Lớp: System (Hệ thống)

Tham chiếu tới: Use case xuất thẻ

Chú ý: Sử dụng phương pháp truy nhập nhanh vào CSDL

Ngoại lệ: Nếu số sê-ri bắt đầu (`tuSeri`) lớn hơn số sê-ri kết thúc (`denSeri`) hoặc các thẻ có sê-ri này không còn thì báo lỗi.

Pre-conditions: Mã đại lý (`maDL`) được biết trước

Post-conditions:

- + Nếu là phiên xuất mới thì tạo ra một đối tượng `Phieuxuat` và kết hợp nó vào `Hethongthe`
- + Tạo ra một đối tượng `Thexuat` cho loại thẻ vừa nhập vào và liên kết nó với `Phieuxuat`.
- + Các giá trị nhập vào được gán cho các thuộc tính của đối tượng `Thexuat` (`tuSeri`, `denSeri`,...)
- + `Thexuat` được liên kết với `Loaithe` dựa vào mã của loại thẻ nhập (`maLoai`)

3.4.2.4 Hợp đồng kết thúc phiên xuất thẻ

Tên gọi: `ketthucXuatthe()`

Trách nhiệm: Kết thúc phiên nhập thẻ, ghi dữ liệu vào CSDL.

Kiểu / Lớp: System (Hệ thống)

Tham chiếu tới: Use case xuất thẻ

Ngoại lệ: Nếu chưa có loại thẻ nào được nhập thì báo lỗi

Post-conditions: Chuyển trạng thái của phiên xuất thẻ sang kết thúc. Cập nhật CSDL

3.4.2.5 Hợp đồng tính dư nợ

Tên gọi: Duno(maDL: text)

Trách nhiệm: Tính tổng số tiền đại lý bán thẻ đang nợ

Kiểu / Lớp: System (Hệ thống)

Tham chiếu tới: Use case thanh toán

Pre-conditions: Mã đại lý (maDL) được biết trước

Post-conditions:

- + Tạo ra một đối tượng Phieuthanhtoan và kết hợp nó vào Hethongthe
- + Tìm trong CSDL các Phieuxuat cho đại lý, tạo ra các đối tượng Phieuxuat, nạp giá trị cho các thuộc tính của chúng và liên kết chúng với Phieuthanhtoan.
- + Với mỗi Phieuxuat, tìm trong CSDL các thẻ xuất tương ứng; tạo ra các đối tượng Thexuat tương ứng, nạp giá trị cho các thuộc tính của Thexuat và liên kết nó với Phieuxuat.
- + Thexuat được liên kết với Loaithe
- + Tạo ra các đối tượng Daily, nạp giá trị cho các thuộc tính của nó và liên kết nó với Phieuthanhtoan.
- + Tính dư nợ của đại lý (tổng tiền đại lý đang nợ) bằng tổng nợ cũ và tổng tiền các hóa đơn chưa thanh toán.

3.4.2.6 Hợp đồng trả bằng tiền mặt

Tên gọi: TrabangTienmat(sotientra: Number)

Trách nhiệm: Ghi nhận thông tin liên quan đến việc thanh toán, tính số tiền dư trả lại đại lý, nếu số dư <0 tiếp tục ghi nợ

Kiểu / Lớp: System (Hệ thống)

Tham chiếu tới: Use case thanh toán

Post-conditions: Tính số tiền dư: `sodu=Phieuthanhtoan.sotientra-Phieuthanhtoan.Tongthanhtoan()`

3.4.2.7 Hợp đồng khởi động hệ thống

Tên gọi: Khoi dong HT ()

Trách nhiệm: Khởi động hệ thống

Kiểu / Lớp: System (Hệ thống)

Post-conditions:

- + Tạo các đối tượng Hethongthe, NhapThe, XuatThe, ThanhToan, Loaithe, Danh mucThe
- + Danh mucThe được liên kết với Loaithe
- + NhapThe, XuatThe được liên kết với Danh mucThe
- + NhapThe, XuatThe, Thanhtoan được liên kết với Hethongthe

3.5 THIẾT KẾ HỆ THỐNG

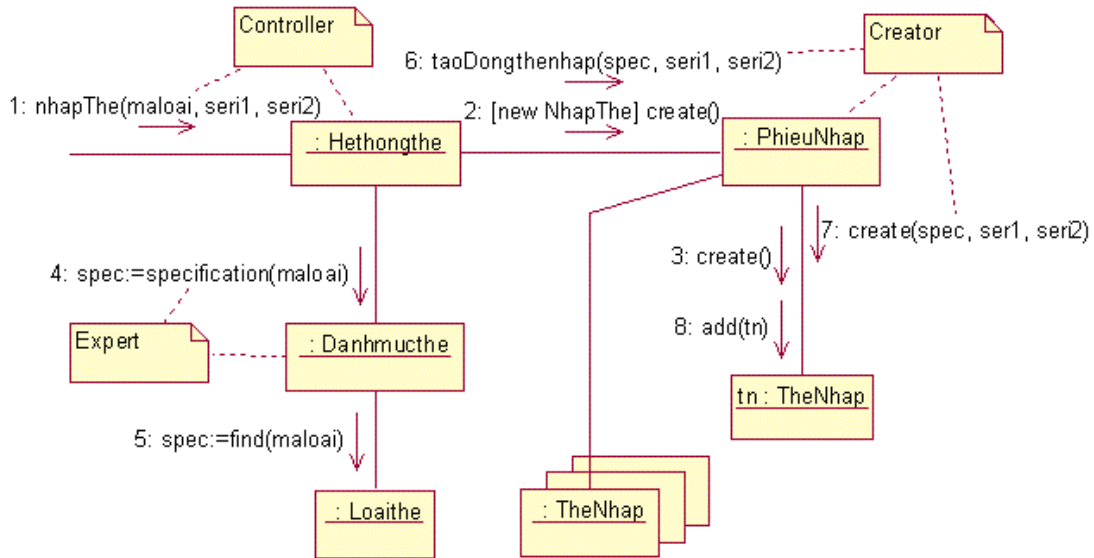
Nhiệm vụ chính của pha thiết kế là xây dựng các biểu đồ cộng tác mô tả chính xác các hoạt động của hệ thống và từ đó thiết kế chi tiết các lớp. Các mẫu GRASP và GoF được lựa chọn sử dụng trong quá trình thiết kế.

3.5.1 CÁC BIỂU ĐỒ CỘNG TÁC

3.5.1.1 Biểu đồ cộng tác cho Hợp đồng nhập thẻ

Theo các thao tác của hợp đồng nhập thẻ, áp dụng các mẫu Controller, Creator, Expert và Low Coupling của GRASP, biểu đồ cộng tác cho hợp đồng nhập thẻ được thiết kế như trong hình 3.13. Khi đó trách nhiệm 1:nhapThe() sẽ được gán cho lớp Hethongthe; trách nhiệm 6:taoDongthenhap() và 7:create() được gán cho lớp PhieuNhap; trách nhiệm lấy thông tin loại thẻ 4:specification() được gán cho lớp Danh mucthe. Ở đây, mẫu Low Coupling được áp dụng trong việc thiết kế để ẩn định trách nhiệm tạo ra *instance* TheNhap

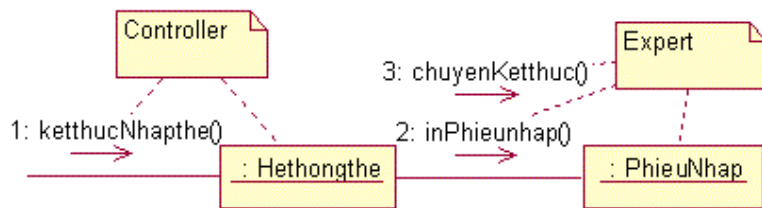
và liên kết nó với PhieuNhap, theo đó trách nhiệm này được gán cho PhieuNhap.



Hình 3.13 Biểu đồ cộng tác hợp đồng nhập thẻ

3.5.1.2 Biểu đồ cộng tác cho Hợp đồng kết thúc phiên nhập thẻ

Biểu đồ cộng tác của hợp đồng kết thúc phiên nhập thẻ được thiết kế như trong hình 3.14.

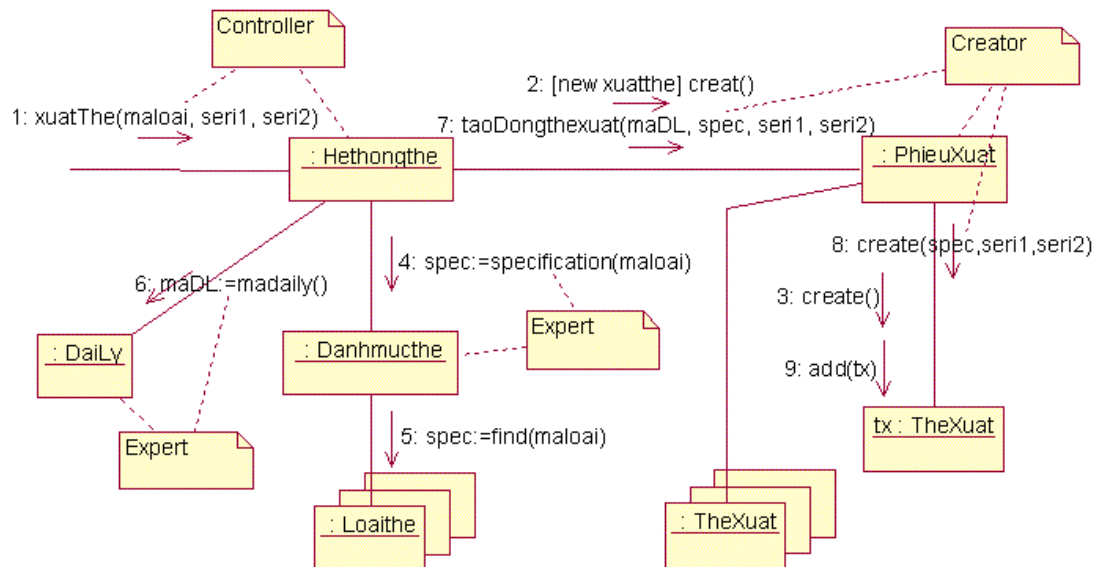


Hình 3.14 Biểu đồ cộng tác hợp đồng kết thúc phiên nhập thẻ

Áp dụng mẫu Controller và mẫu Expert trong GRASP, trách nhiệm ketthucNhapthe() sẽ được gán cho lớp Hethongthe, trách nhiệm inPhieunhap() và chuyenKetthuc() sẽ được gán cho lớp PhieuNhap.

3.5.1.3 Biểu đồ cộng tác cho Hợp đồng xuất thẻ

Tương tự như trong hợp đồng nhập thẻ, biểu đồ cộng tác của hợp đồng xuất thẻ được thiết kế như trong hình 3.15. Trong đó trách nhiệm 1:xuatThe() sẽ được gán cho lớp Hethongthe; trách nhiệm 7:taoDongthexuat() và 8:create() được gán cho lớp PhieuXuat; trách nhiệm lấy thông tin loại thẻ 4:specification() được gán cho lớp Danh muc the và trách nhiệm lấy mã đại lý 6:madaily() được gán cho lớp DaiLy. Trách nhiệm tạo ra *instance* TheXuat và liên kết nó với PhieuXuat, được gán cho chính PhieuXuat theo mẫu Low Coupling.

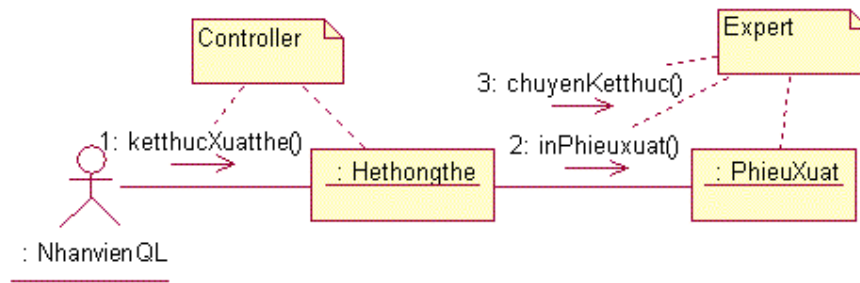


Hình 3.15 Biểu đồ cộng tác xuất thẻ

3.5.1.4 Biểu đồ cộng tác cho Hợp đồng kết thúc phiên xuất thẻ

Biểu đồ cộng tác của hợp đồng kết thúc phiên xuất thẻ được thể hiện trong hình 3.16.

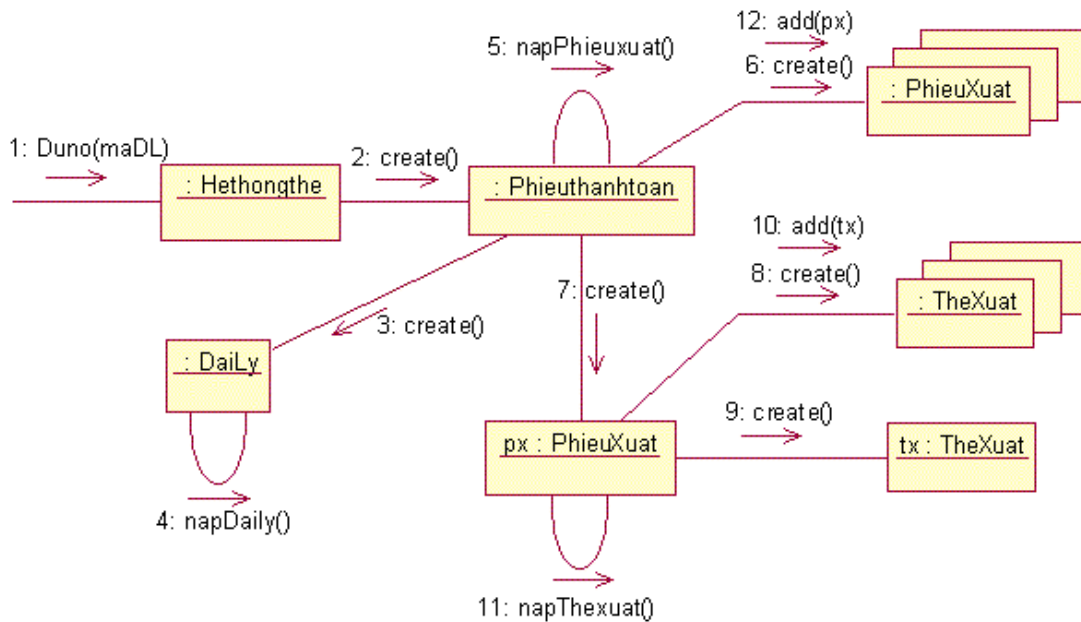
Áp dụng mẫu Controller và mẫu Expert trong GRASP, trách nhiệm ketthucXuatthe() được gán cho lớp Hethongthe, trách nhiệm inPhieuxuat() và chuyenKetthuc() được gán cho lớp PhieuXuat.



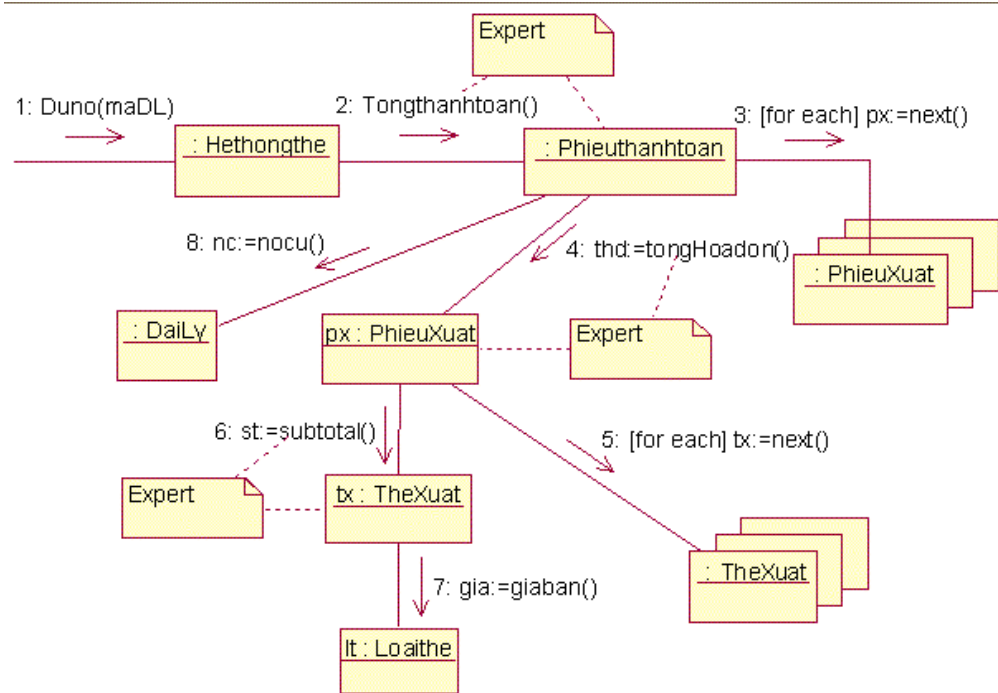
Hình 3.16 Biểu đồ cộng tác hợp đồng kết thúc phiên xuất thẻ

3.5.1.5 Biểu đồ cộng tác cho Hợp đồng tính dư nợ của đại lý

Hợp đồng tính dư nợ thực chất là tính tổng tiền mà đại lý phải thanh toán. Tổng tiền thanh toán bằng tổng giá trị các đợt xuất thẻ cho đại lý và số tiền đại lý còn nợ của lần thanh toán trước. Biểu đồ cộng tác tính dư nợ được thể hiện thông qua hai biểu đồ cộng tác thành phần được thể hiện trong hình 3.17 và hình 3.18.



Hình 3.17 Biểu đồ cộng tác khởi tạo dư nợ



Hình 3.18 Biểu đồ cộng tác tính dư nợ

Để thực hiện công việc thanh toán tiền của một đại lý bán thẻ, đối tượng điều khiển phải có khả năng lấy được tất cả các hợp đồng xuất thẻ cho đại lý kể từ lần thanh toán cuối cùng đến thời điểm thanh toán, vì vậy áp dụng mẫu điều khiển - GRASP Controller, hợp đồng tính dư nợ Duno(maDL) sẽ được gán cho lớp điều khiển Hethongthe.

Áp dụng mẫu chuyên gia - GRASP Expert, trách nhiệm gán cho các lớp Thanhtoan, Xuatthe, TheXuat, Loaithe, Daily như trong bảng 3.2.

Lớp thiết kế	Trách nhiệm/nhiệm vụ
Phieuthanhtoan	Biết tổng tiền nợ của đại lý (Tongthanhtoan(maDL))
Xuatthe	Biết tổng tiền của từng hóa đơn xuất (tongHoadon())
TheXuat	Biết tổng tiền của từng loại loại thẻ được xuất (subtotal())
Loaithe	Biết giá bán cho đại lý của loại thẻ (giaban())
Daily	Biết số tiền đại lý còn nợ của lần thanh toán trước (nocu())

Bảng 3.2 Trách nhiệm gán cho các lớp Phieuthanhtoan, Xuatthe, TheXuat, Loaithe, Daily

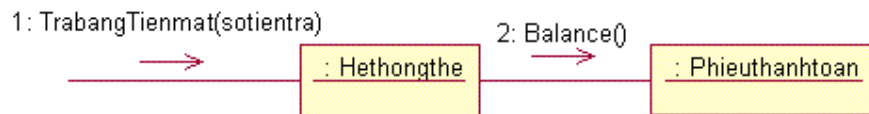
Khi đó các hàm tổng tongtien(), tongHoadon() và subtotal() được thực hiện như sau:

```
Phieuthanhtoan--Tongthanhtoan()
{
    ttien:=0
    for each PhieuXuat, px
        ttien:=ttien + px.tongHoadon()
    return ttien + Daily.nocu()
}
```

```
PhieuXuat--tongHoadon()
{
    thd:=0
    for each TheXuat, tx
        thd:=thd + tx.subtotal()
    return thd
}
```

```
TheXuat--subtotal()
{
    return soluong*lt.giaban()
}
```

3.5.1.6 Biểu đồ cộng tác cho Hợp đồng thanh toán bằng tiền mặt



Hình 3.19 Biểu đồ cộng tác tính dư nợ

Biểu đồ công tác hợp đồng trả bằng tiền mặt được thể hiện trong hình 3.19. Một công việc trong việc thanh toán bằng tiền mặt là tính tiền chênh lệch giữa số tiền mặt trả và tổng số nợ của đại lý (Balance()). Phần công việc này sẽ được gán trách nhiệm cho lớp Phieuthanhtoan.

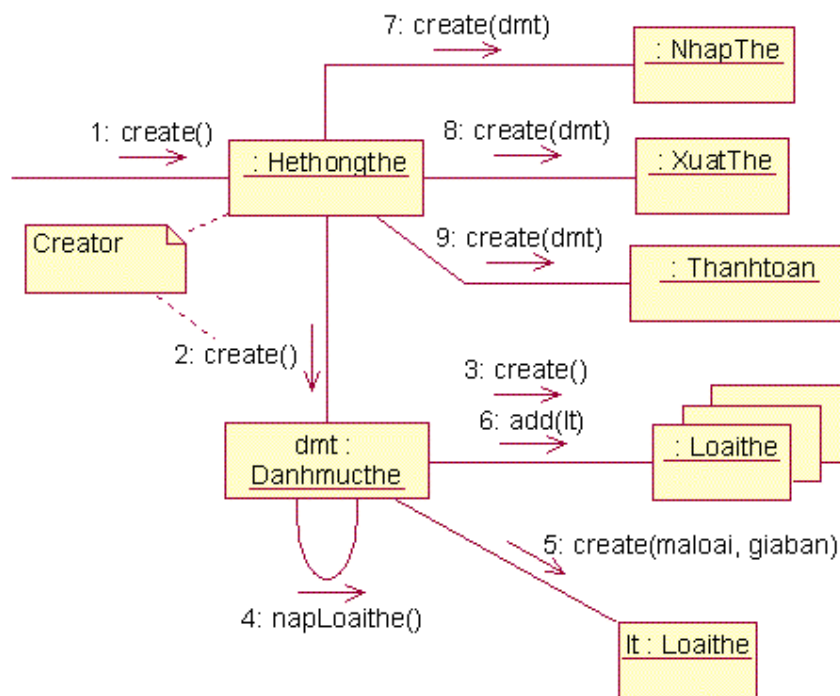
```
Phieuthanhtoan--Balance()
{
    return sotientra - Tongthanhtoan();
}
```

3.5.1.7 Biểu đồ cộng tác cho Hợp đồng khởi động hệ thống

Việc khởi động hệ thống là việc đầu tiên hệ thống phải thực hiện, nhưng khi thiết kế, hợp đồng này được thiết kế sau để đảm bảo mọi thông tin liên quan đến các hoạt động sau này của hệ thống đều đã được xác định.

Biểu đồ cộng tác của Khoi dong HT chỉ ra những gì có thể xảy ra khi đối tượng của miền ứng dụng được tạo ra. Nghĩa là trong hệ thống phải gán trách nhiệm `create()` cho lớp chính `Hethongthe`.

Căn cứ vào hợp đồng của Khoi dong HT và mẫu GRASP, biểu đồ cộng tác cho Khoi dong HT như hình 3.20

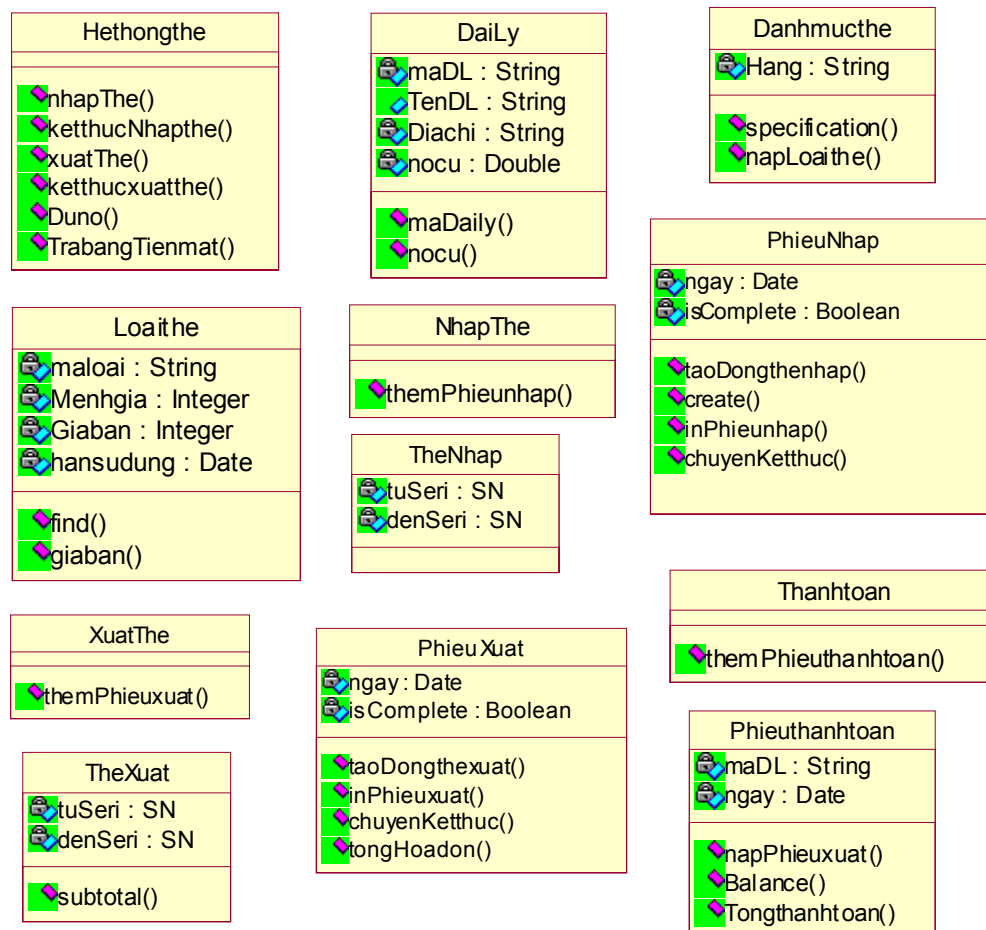


Hình 3.20 Biểu đồ cộng tác khởi động hệ thống

3.5.2 BIỂU ĐỒ LỚP THIẾT KẾ

3.5.2.1 Thiết kế lớp

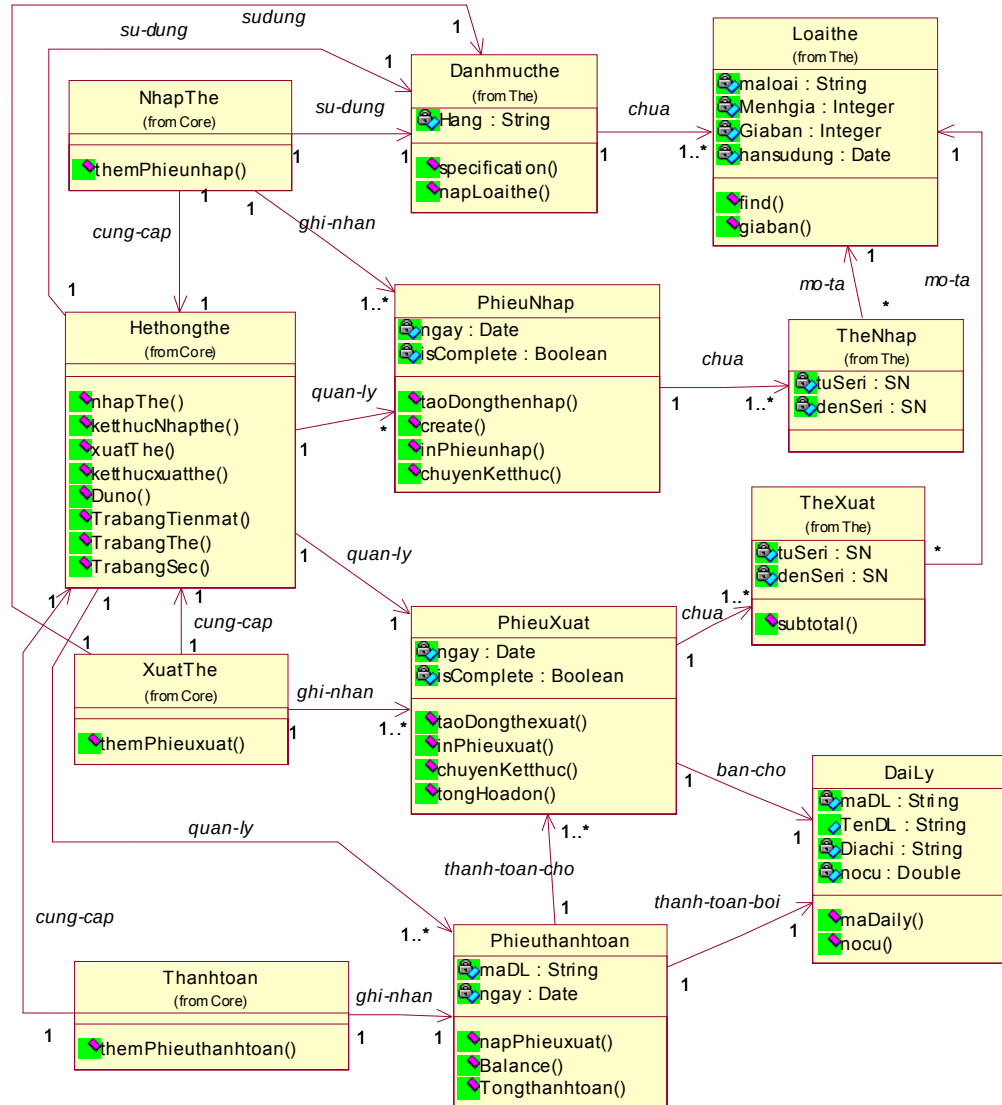
Các lớp thiết kế được thiết kế dựa trên việc biến đổi lớp phân tích (từ mô hình khái niệm) thành lớp thiết kế. Tiếp theo bổ sung các phương thức hay các thao tác và kiểu dữ liệu của chúng cho các lớp dựa theo thiết kế của các biểu đồ cộng tác. Từ mô hình khái niệm và các biểu đồ cộng tác theo thiết kế trên, các lớp thiết kế cơ bản của hệ thống quản lý thẻ được thể hiện như trong hình 3.21.



Hình 3.21 Các lớp thiết kế cơ bản của hệ thống

3.5.2.2 Bổ sung quan hệ giữa các lớp

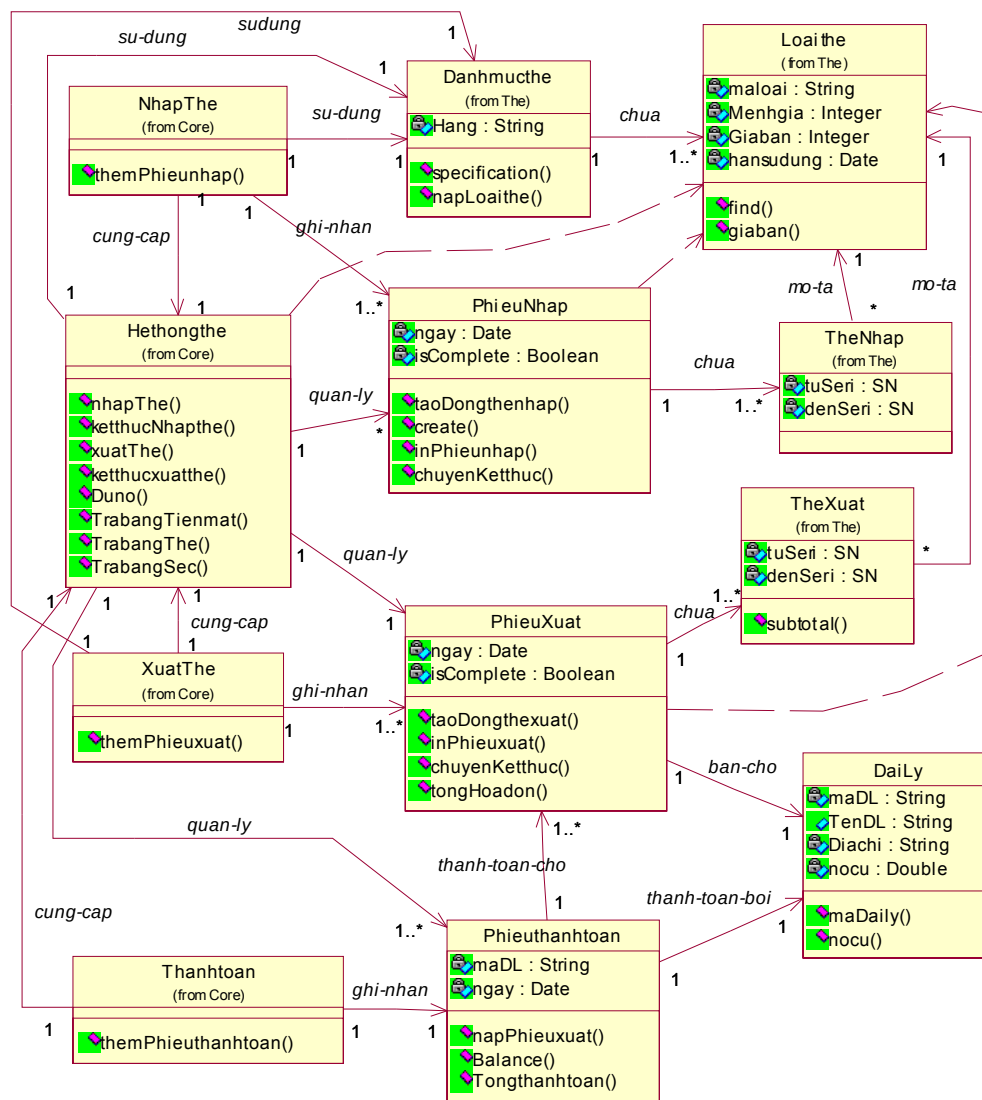
Thiết kế biểu đồ lớp là việc xác định mối quan hệ giữa các lớp. Mỗi quan hệ giữa các lớp được biểu diễn bằng một mũi tên từ lớp nguồn đến lớp đích. Trên mũi tên có các thông tin xác định quan hệ và tên mối quan hệ.



Hình 3.22 Biểu đồ lớp thiết kế của hệ thống

Mối quan hệ giữa các lớp được xác định trên cơ sở mối quan hệ trong mô hình khái niệm và quá trình tương tác giữa các đối tượng trong các mô hình cộng tác. Thông thường mối quan hệ giữa hai lớp A và B được xác lập khi A gửi thông điệp cho B và/hoặc A tạo ra một *instance* B hoặc A cần duy trì sự liên kết với B. Biểu đồ lớp của hệ thống quản lý thẻ được xây dựng và thể hiện như trong hình 3.22.

3.5.2.3 Bổ sung quan hệ phụ thuộc



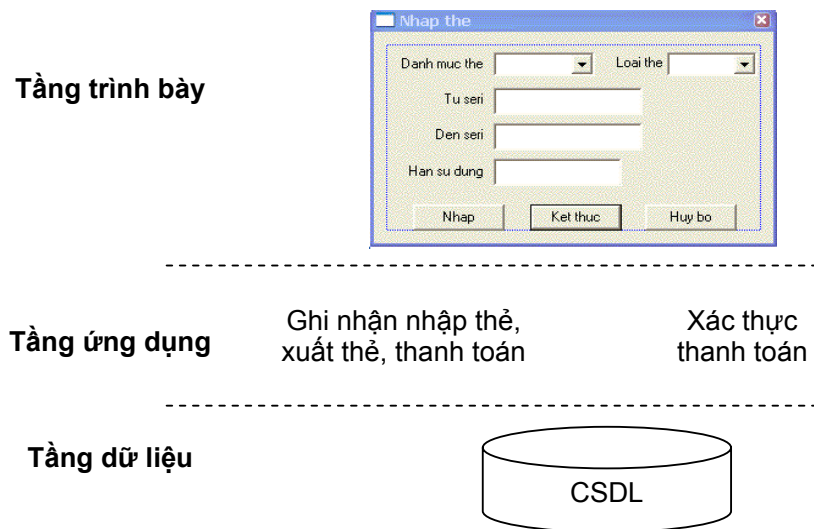
Hình 3.23 Biểu đồ lớp thiết kế bổ sung quan hệ phụ thuộc của hệ thống

Quan hệ phụ thuộc chỉ ra việc một thành phần biết một thành phần khác. Trong biểu đồ lớp quan hệ phụ thuộc rất hiệu quả để mô tả khả năng ‘nhìn thấy’ giữa các lớp. Khả năng nhìn thấy được của một đối tượng gồm: tầm nhìn thuộc tính, tầm nhìn tham số, tầm nhìn khai báo cục bộ và tầm nhìn toàn cục.

Trong hệ thống quản lý thẻ lớp Hethongthe tiếp nhận đối tượng trả lại của kiểu Loaithe từ một thông điệp mà nó gửi cho Danhmucthe, do đó Hethongthe có khả năng nhìn thấy Loaithe. Lớp PhieuNhap và PhieuXuat tiếp nhận Loaithe như một tham số trong phương thức taoDongthenhap() và taoDongthexuat(). Như vậy Hethongthe, PhieuNhap, PhieuXuat có quan hệ phụ thuộc tới Loaithe. Bổ sung các quan hệ phụ thuộc, biểu đồ lớp thiết kế của hệ thống được thể hiện như hình 3.23.

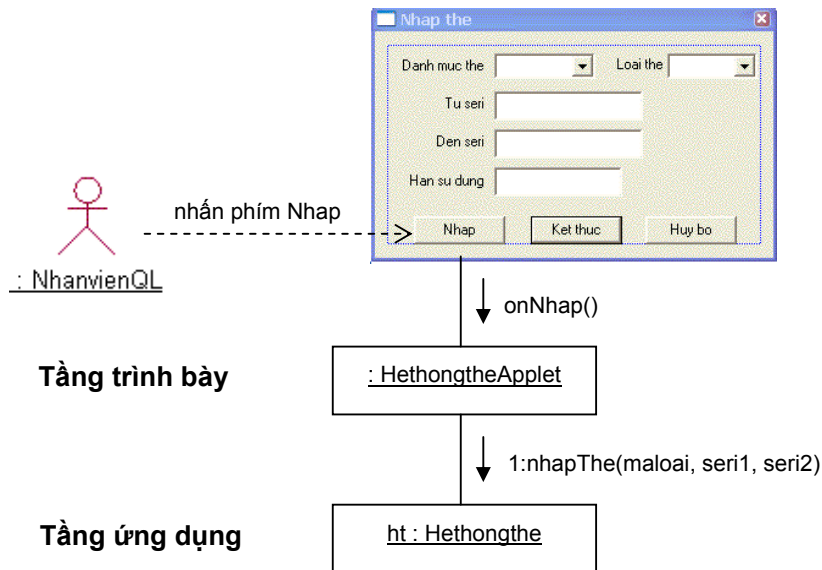
3.5.3 THIẾT KẾ TRIỂN KHAI

Hệ thống quản lý thẻ điện thoại được thiết kế triển khai theo kiến trúc ba



Hình 3.24 Kiến trúc ba tầng của hệ thống
tầng bao gồm tầng trình bày, tầng ứng dụng và tầng lưu trữ như trong hình 3.24

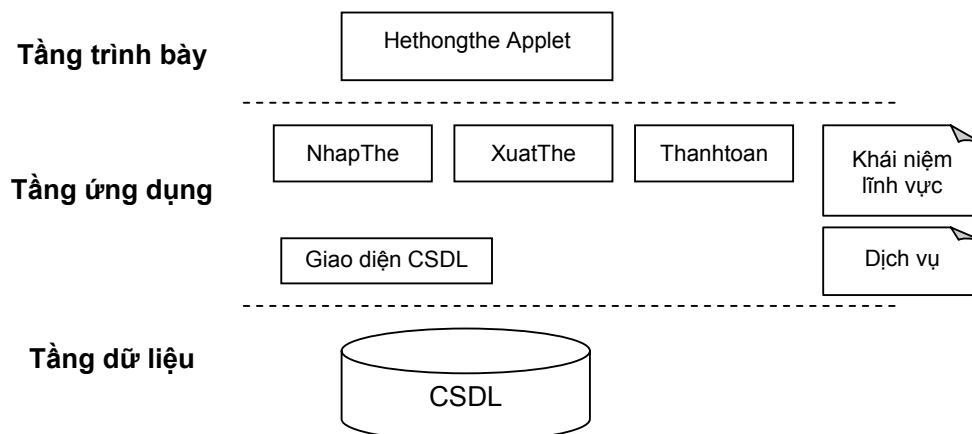
Mối liên hệ giữa tầng trình bày và tầng ứng dụng được thực hiện thông qua lớp HethongtheApplet như ví dụ trong hình 3.25.



Hình 3.25 Liên kết giữa tầng trình bày và tầng ứng dụng

Hệ thống tiếp tục được phân chia thành các lớp mịn hơn như hình 3.26. Tầng ứng dụng lúc này bao gồm các lớp sau:

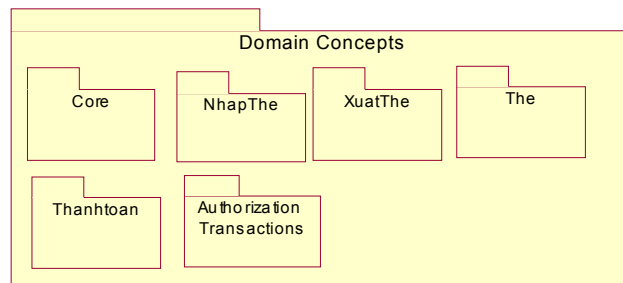
- Domain Objects - Các lớp biểu diễn các khái niệm lĩnh vực, ví dụ: NhapThe, XuatThe, Thanhtoan
- Services - Các đối tượng phục vụ cho các hàm như trao đổi với CSDL, báo cáo, liên kết, an toàn,...



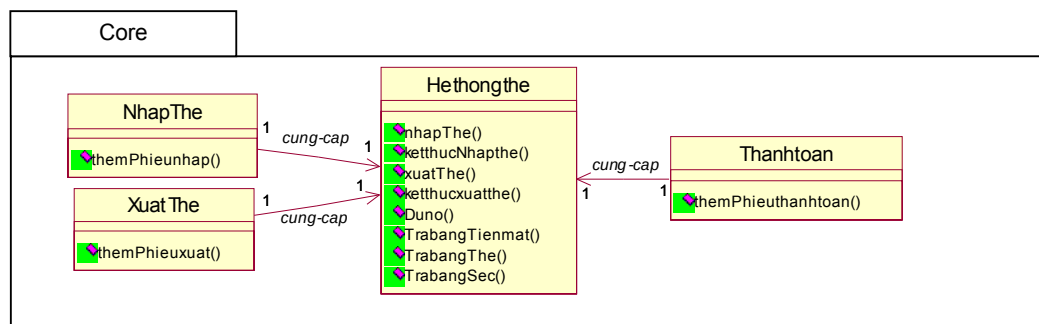
Hình 3.26 Kiến trúc ba tầng của hệ thống

Gói thiết kế

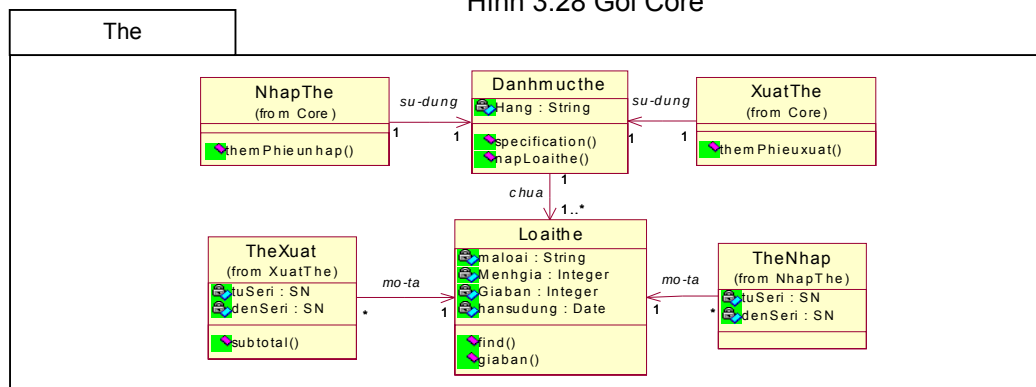
Các lớp trong Hệ thống quản lý thẻ được tổ chức lại thành các gói khái niệm (package) nhằm làm cho việc quan sát mô hình thiết kế trở nên đơn giản hơn. Các gói khái niệm của Hệ thống quản lý thẻ được thiết kế bao gồm: Domain Concepts, Core, NhậpThe, XuấtThe, The, Thanhtoan (Hình 3.27 - hình 3.32)



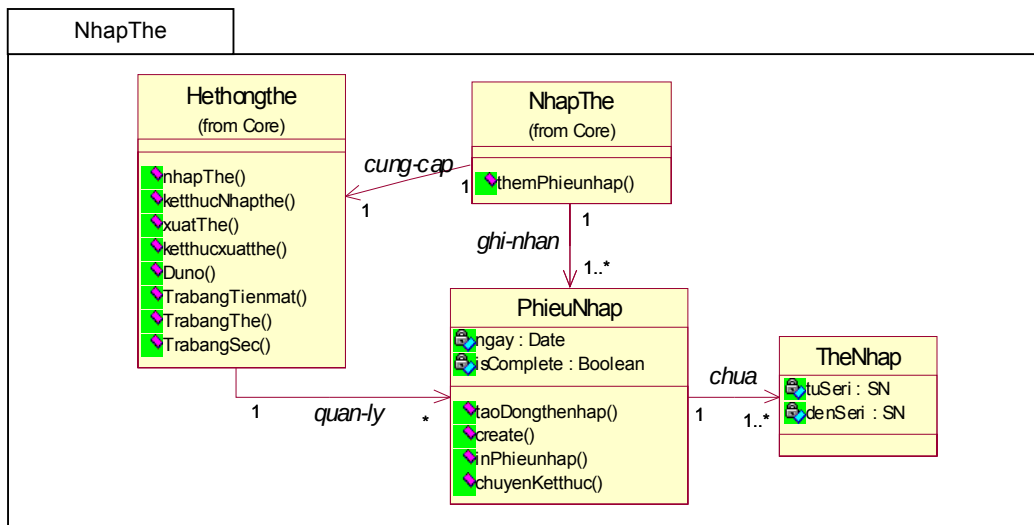
Hình 3.27 Gói Domain Concepts



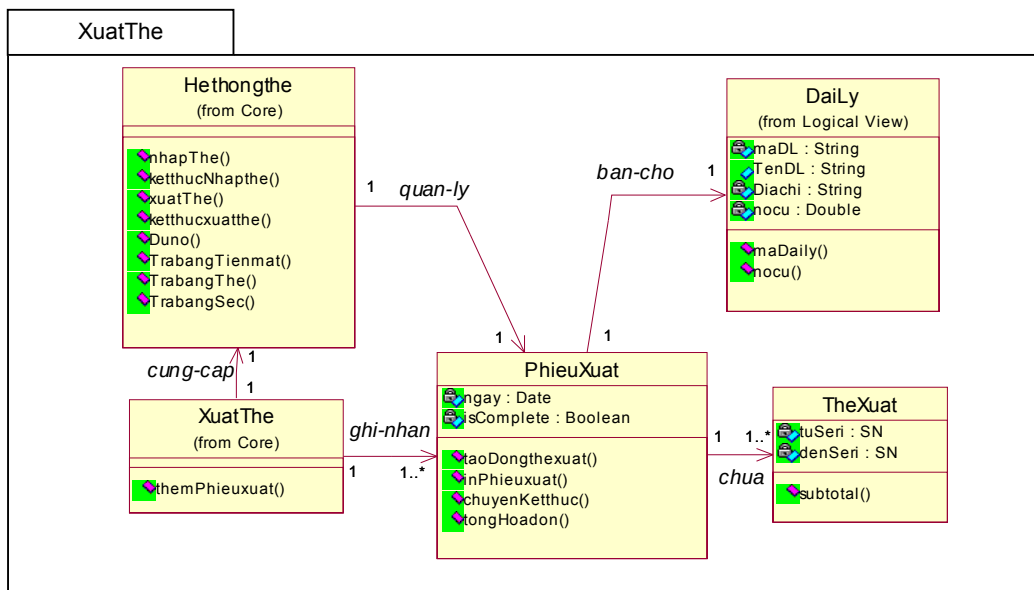
Hình 3.28 Gói Core



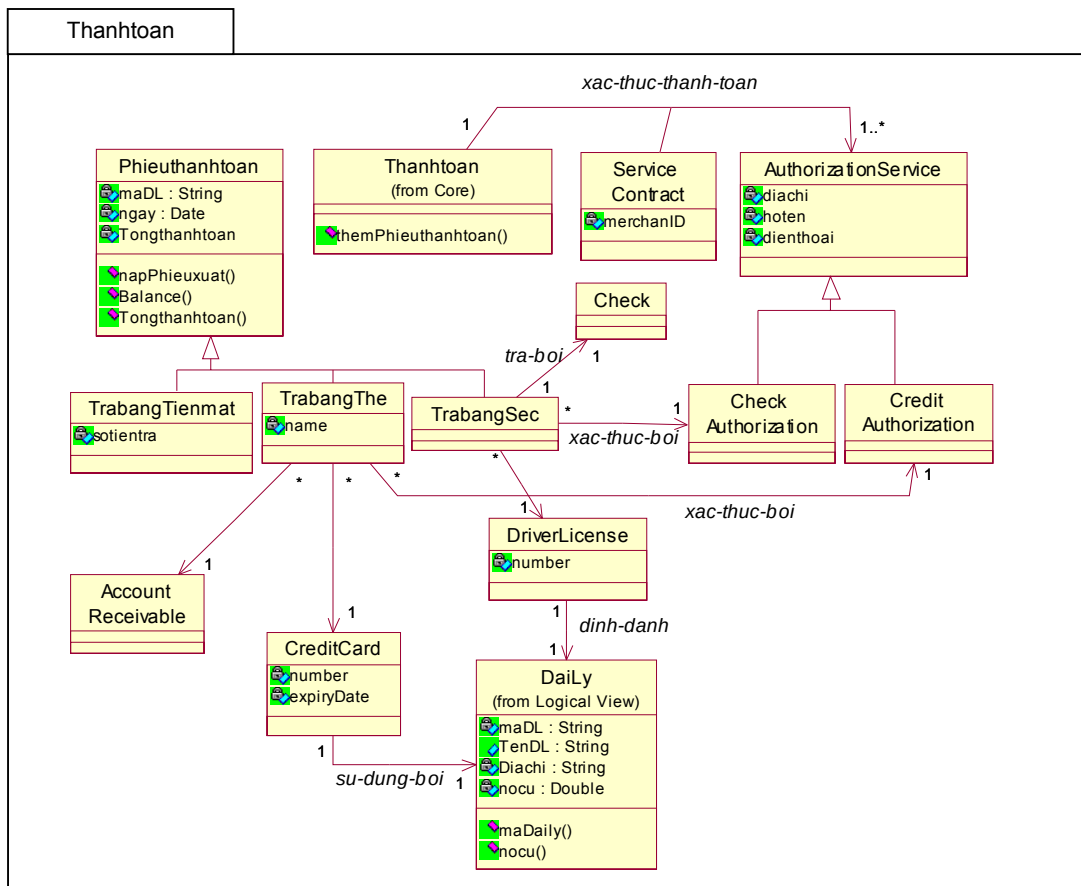
Hình 3.29 Gói The



Hình 3.30 Gói NhapThe



Hình 3.31 Gói XuatThe



Hình 3.32 Gói Thanhtoan

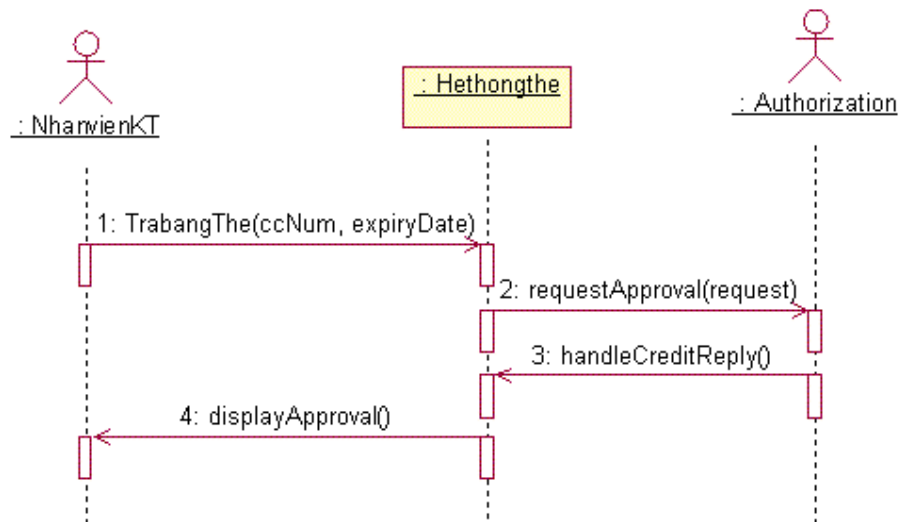
Ở đây hệ thống được bổ sung khả năng cho phép đại lý thanh toán (trả) bằng thẻ tín dụng và séc. Do đó hệ thống được thiết kế bổ sung các lớp TrabangTienmat, TrabangThe, TrabangSec, AccountReceivable, CreditCard, DriverLicense, Check, CheckAuthorization, CreditAuthorization, AuthorizationService, và ServiceContract.

3.5.4 BỔ SUNG THIẾT KẾ

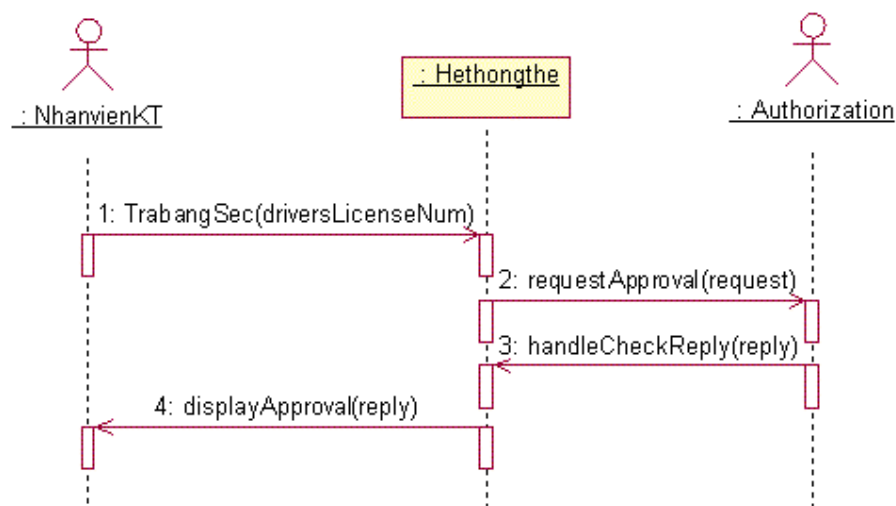
3.5.4.1 Bổ sung biểu đồ trình tự thanh toán bằng thẻ tín dụng và séc

Khi thanh toán, đại lý có thể trả bằng thẻ tín dụng. Thẻ tín dụng được xác định thông qua số hiệu thẻ (ccNum) và hạn sử dụng (expiryDate). Điều đó được thể

hiện bằng việc gửi một thông điệp `TrabangThe(ccNum, expiryDate)` cho hệ thống. Hệ thống gửi yêu cầu kiểm duyệt thẻ `requestApproval(request)` tới bộ phận kiểm duyệt thẻ tín dụng là tác nhân ngoài của hệ thống. Hệ thống sẽ dựa vào kết quả trả lời của bộ phận kiểm duyệt để thanh toán với khách hàng bằng cách trừ đi số tiền mua hàng trong thẻ tín dụng nếu nó hợp pháp, nghĩa là thực hiện `handleCreditReply()` (xử lý thẻ đã kiểm duyệt), dựa vào kết quả của bộ phận kiểm duyệt. Biểu đồ trình tự mô tả sự tương tác giữa người tiếp đón, hệ thống và bộ phận kiểm duyệt thẻ `Authorization` được thiết lập như hình 3.33.



Hình 3.33 Biểu đồ trình tự thanh toán bằng thẻ tín dụng



Hình 3.34 Biểu đồ trình tự thanh toán bằng séc

Tương tự, ta có thể thiết kế biểu đồ trình tự của thành toán bằng séc như hình 3.34.

3.5.4.2 Bổ sung các hợp đồng thanh toán bằng thẻ tín dụng và séc

Hợp đồng thanh toán bằng thẻ tín dụng

<i>Tên gọi:</i>	TrabangThe(ccNum, expiryDate)
<i>Trách nhiệm:</i>	Tạo và yêu cầu chứng thực cho việc thanh toán bằng thẻ
<i>Kiểu / Lớp:</i>	System (Hệ thống)
<i>Tham chiếu tới:</i>	Use case Thanh toán
<i>Kết quả:</i>	Một yêu cầu thanh toán được gửi đến cho bộ phận kiểm duyệt thẻ
<i>Post-conditions:</i>	+ Một thanh toán thẻ pmt được tạo ra. + pmt sẽ liên kết với phiên thanh toán hiện tại. + Một thẻ cc được tạo ra, cc.number = ccNum, cc.expiryDate = expiryDate. + cc liên kết với pmt. + Một yêu cầu thanh toán thẻ cpr được tạo ra. + pmt liên kết với cpr. + cpr liên kết với dịch vụ xác thực thẻ CreditAuthorization.

Hợp đồng thao tác phản hồi kiểm tra thẻ

<i>Tên gọi:</i>	HandleCreditReply(reply: CreditPaymentReply)
<i>Trách nhiệm:</i>	Phản hồi trả lời kiểm duyệt từ bộ phận kiểm duyệt. Nếu chấp nhận thì hoàn thành và ghi nhận thanh toán
<i>Kiểu / Lớp:</i>	System (Hệ thống)
<i>Tham chiếu tới:</i>	Use case Thanh toán
<i>Kết quả:</i>	Nếu chấp nhận, một phản hồi chấp nhận thanh toán bằng thẻ (creditPaymentApprovalReply) được gửi đến cho bộ phận tiếp nhận tài khoản (AccountReceivable)

Post-conditions: Nếu trả lời là chấp nhận thì:
 + Một trả lời chấp nhận thanh toán bằng thẻ approval được tạo ra.
 + approval liên kết với AccountReceivable.
 Nếu không chấp nhận:
 + Một trả lời từ chối thanh toán bằng thẻ denial được tạo ra.

Hợp đồng thanh toán bằng bằng séc

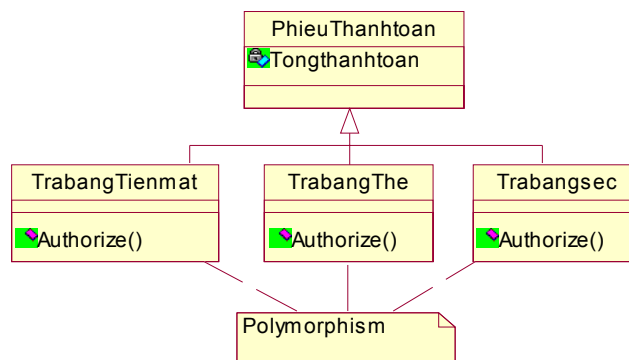
Tên gọi: TrabangSec(driversLicenceNum)
Trách nhiệm: Tạo và yêu cầu chứng thực cho việc thanh toán bằng séc.
Kiểu / Lớp: System (Hệ thống)
Tham chiếu tới: Use case Thanh toán
Tiền điều kiện Phiên tiếp đón hiện tại hoàn thành
Post-conditions: + Một thanh toán bằng séc pmt được tạo ra
 + pmt sẽ liên kết với phiên thanh toán hiện tại
 + Một *drivers License* dl được tạo ra, dl.number = driversLicenseNum.
 + dl liên kết với pmt.
 + Một yêu cầu thanh toán bằng séc cpr được tạo ra.
 + pmt liên kết với cpr.
 + cpr liên kết với CreditAuthorizationService.

Hợp đồng thao tác phản hồi kiểm tra séc

Tên gọi: HandleCheckReply(reply: CheckPaymentReply)
Trách nhiệm: Phản hồi trả lời kiểm duyệt từ bộ phận kiểm duyệt. Nếu chấp nhận thì hoàn thành phiên thanh toán và ghi nhận thanh toán
Kiểu / Lớp: System (Hệ thống)
Tham chiếu tới: Use case Thanh toán
Tiền điều kiện: Yêu cầu kiểm tra séc được gửi đến bộ phận kiểm tra séc

Post-conditions: Nếu trả lời là chấp nhận thì:
 + Một trả lời chấp nhận thanh toán bằng séc approval được tạo ra
 Nếu không chấp nhận:
 + Một trả lời từ chối thanh toán bằng séc denial được tạo ra

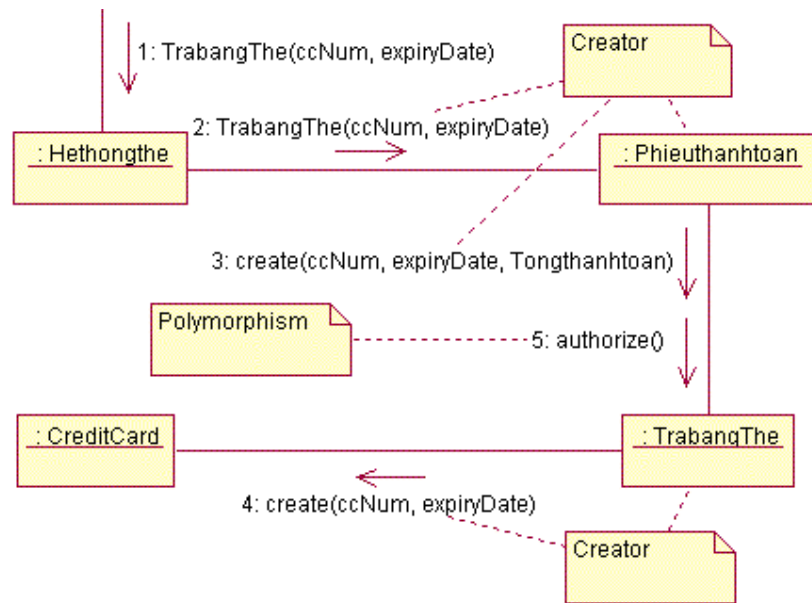
Khi thực hiện hợp đồng TrabangThe, TrabangSec một công việc chung cần thực hiện là kiểm tra xác thực - `Authorize()`. Áp dụng mẫu Polymorphism của GRASP, các lớp phục vụ thanh toán trong gói Thanhtoan được thiết kế như hình 3.35. Mỗi hình thức thanh toán phải tự mình thực hiện việc kiểm tra xác thực khi thanh toán.



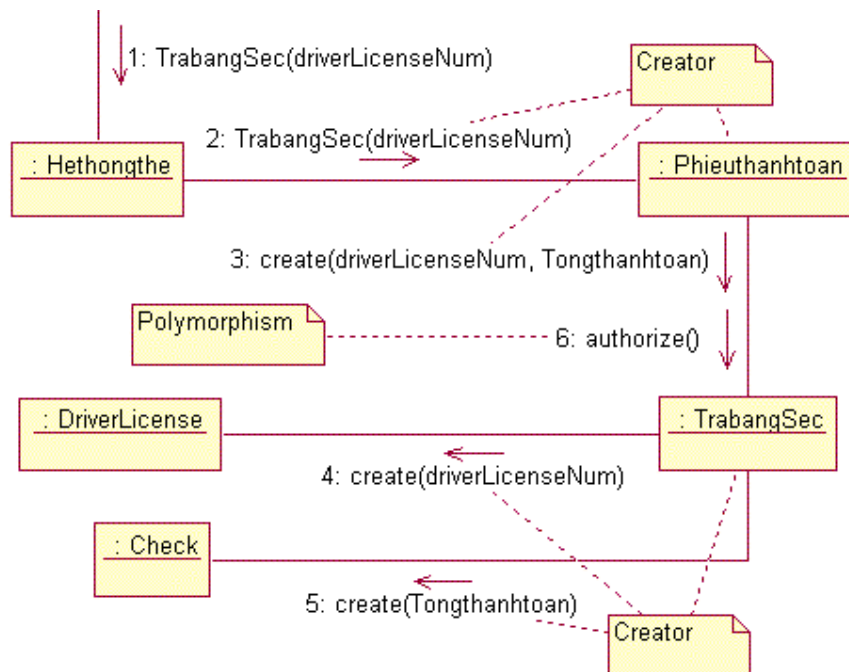
Hình 3.35 Polymorphism trong xác thực thanh toán

3.5.4.3 Bổ sung biểu đồ cộng tác

Biểu đồ cộng tác khởi tạo hợp đồng TrabangThe và hợp đồng TrabangSec được thiết kế như hình 3.36 và hình 3.37. Mẫu Creator và Polymorphism được áp dụng trong quá trình thiết kế để gán trách nhiệm cho các lớp.



Hình 3.36 Biểu đồ cộng tác khởi tạo hợp đồng TrabangThe

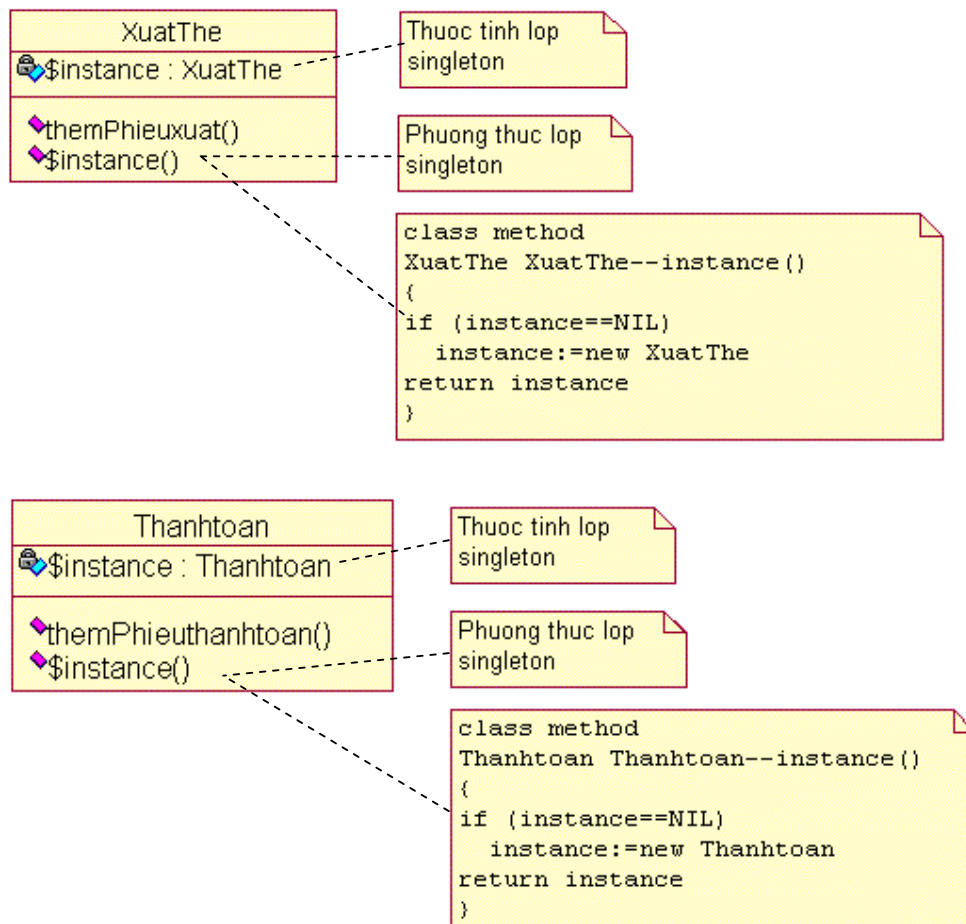


Hình 3.37 Biểu đồ cộng tác khởi tạo hợp đồng TrabangSec

3.5.4.4 Áp dụng thêm các mẫu trong thiết kế

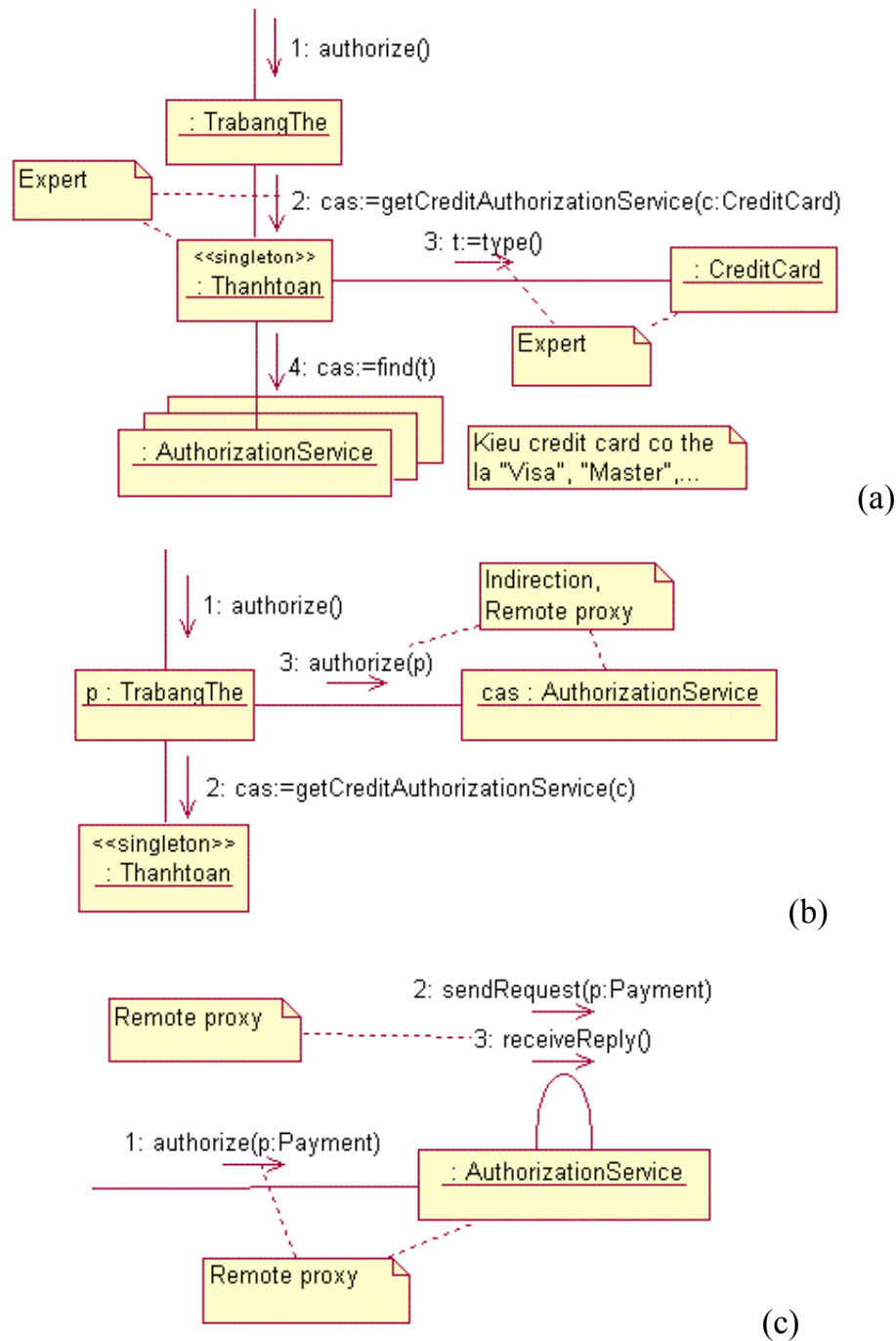
i) Áp dụng mẫu GoF Singleton

Để bảo đảm trong suốt quá trình làm việc của hệ thống, các lớp XuatThe và Thanhtoan chỉ có thể tạo ra một thể hiện duy nhất để tránh việc có nhiều phiên xuất thẻ cùng xuất cho một đại lý hoặc có nhiều phiên thanh toán cùng thực hiện cho một đại lý, mẫu GoF Singleton được áp dụng cho việc thiết kế các lớp này (Hình 3.38).



Hình 3.38 Áp dụng Mẫu Singleton thiết kế lớp XuatThe và Thanhtoan

ii) Áp dụng mẫu GoF Remote Proxy và Proxy



Hình 3.39 Áp dụng mẫu Remote proxy: (a) tìm dịch vụ xác thực, (b) sử dụng Remote proxy và (c) các hành động Remote proxy

Trong việc thanh toán bằng thẻ tín dụng, hệ thống cần phải giao dịch với một dịch vụ xác thực thẻ bên ngoài - `CreditAuthorizationService`. Trong

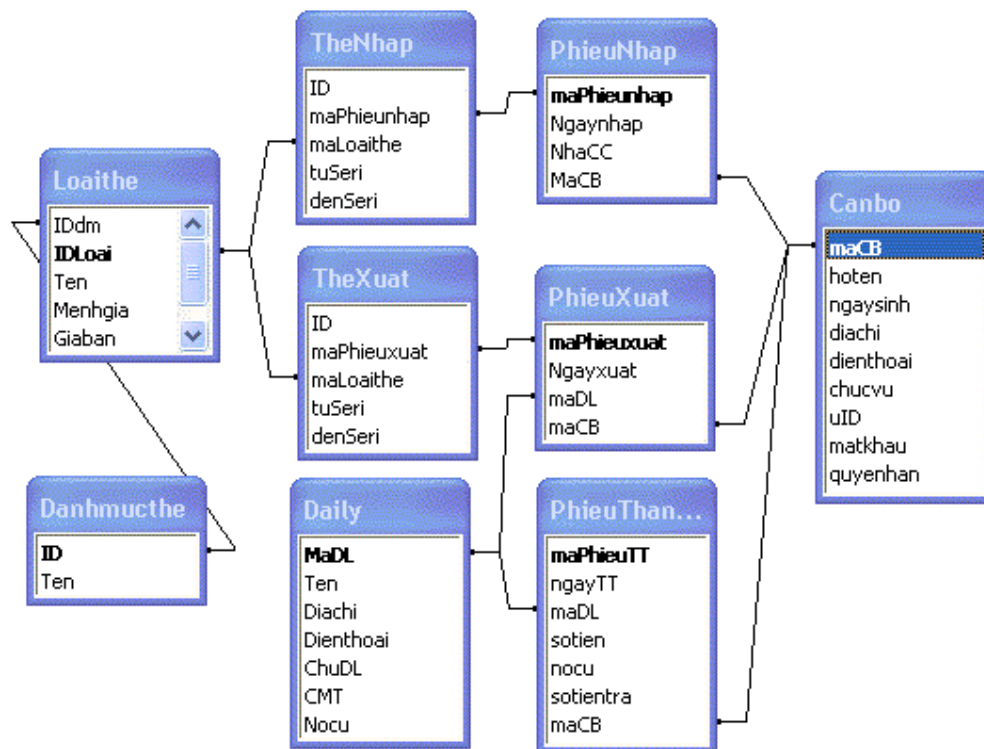
trường hợp này, theo mẫu GoF Remote Proxy, ta có thể tạo ra một lớp nội bộ làm đại diện liên lạc với dịch vụ thực và đối tượng CreditAuthorizationService được sử dụng để giao tiếp với bên ngoài.

Khi thực hiện thanh toán bằng thẻ, hệ thống cần tìm dịch vụ xác thực thẻ, sau đó trên cơ sở remote proxy mới đề nghị dịch vụ này giao dịch với dịch vụ thực tế.

Hình 3.39 thể hiện các biểu đồ cộng tác thực hiện việc tìm dịch vụ khi thanh toán bằng thẻ tín dụng, sử dụng mẫu Remote proxy và các hành động của nó.

3.5.5 MÔ HÌNH HÓA DỮ LIỆU

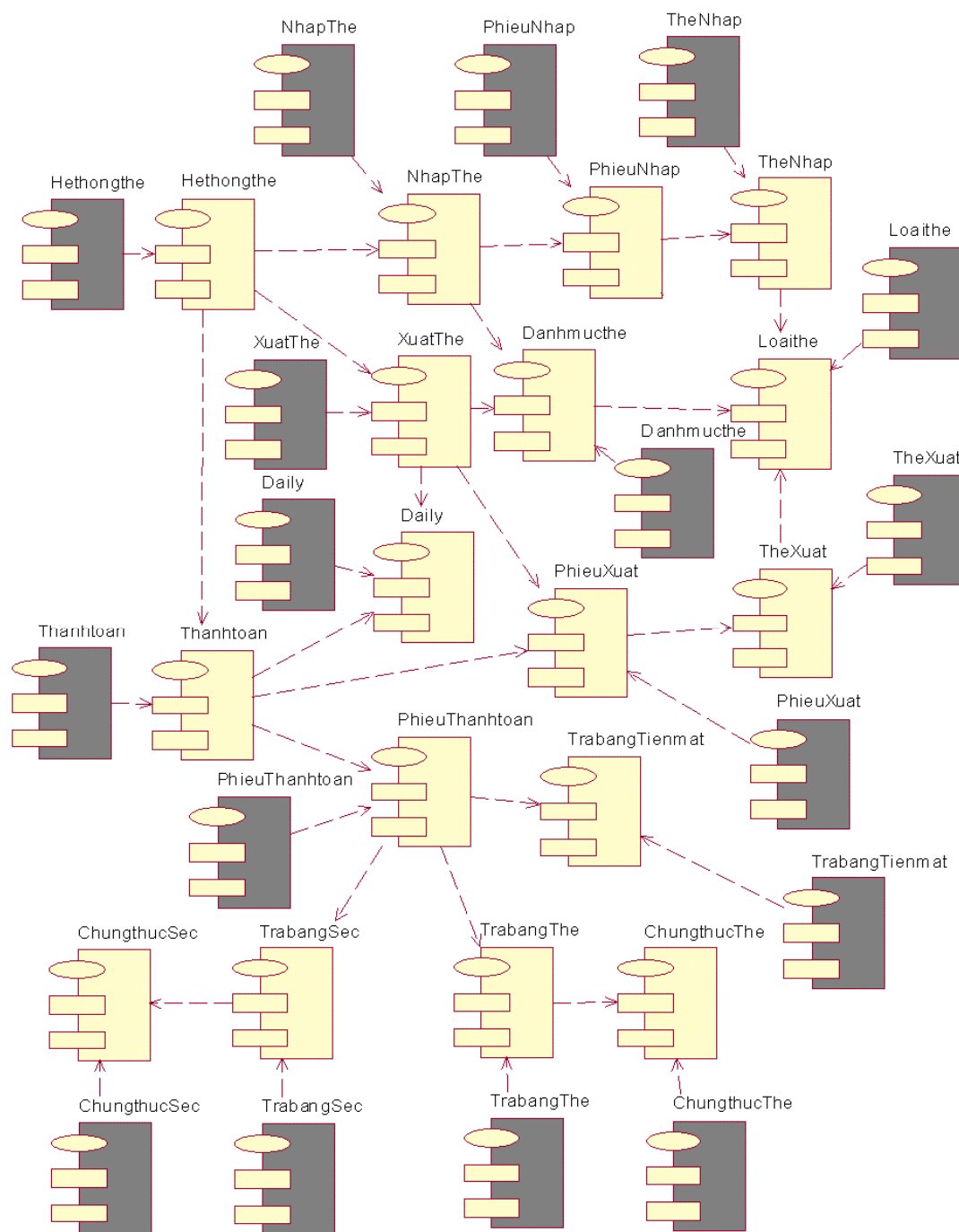
Trên cơ sở phân tích và thiết kế, phần chính cơ sở dữ liệu của Hệ thống quản lý thẻ được thiết kế như hình 3.40.



Hình 3.40 Biểu đồ quan hệ dữ liệu của hệ thống

3.6 CÀI ĐẶT THIẾT KẾ

3.6.1 BIỂU ĐỒ THÀNH PHẦN

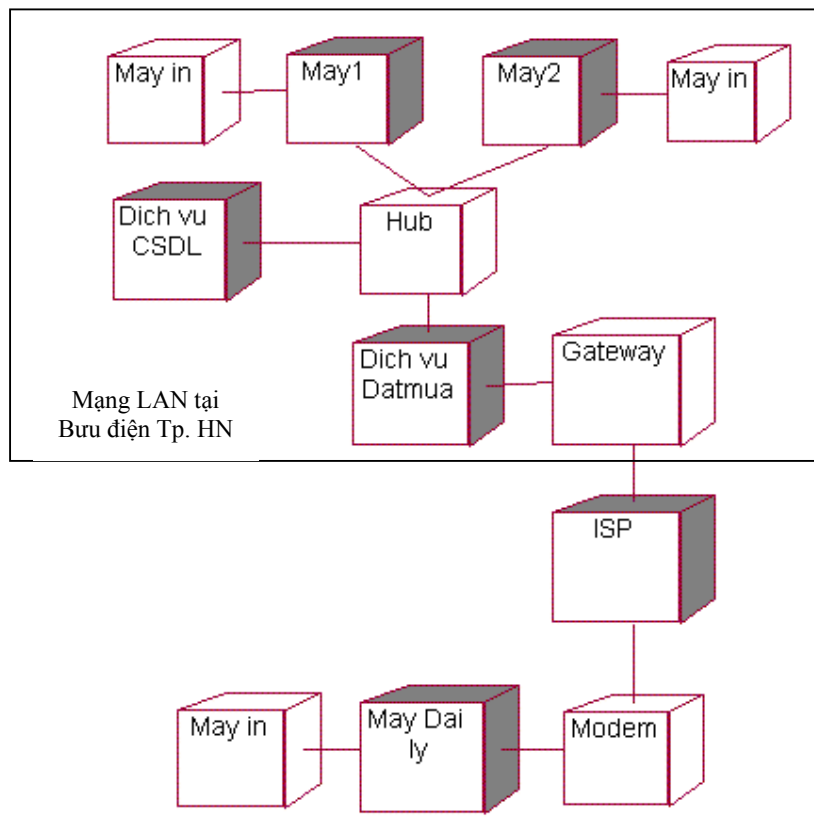


Hình 3.41 Biểu đồ thành phần Hệ thống quản lý thẻ

Thành phần là mô đun vật lý mã trình, thành phần phần mềm có thể là thư viện mã nguồn và các tệp chạy được. Mặc định mỗi lớp thiết kế sẽ có phần đặc tả và phần thân. Đặc tả chứa ghép nối lớp, thân chứa cài đặt của cùng lớp đó. Biểu đồ thành phần thể hiện các thành phần của hệ thống và phụ thuộc giữa chúng. Biểu đồ thành phần hệ thống quản lý thẻ được thiết kế như hình 3.41.

3.6.2 BIỂU ĐỒ TRIỂN KHAI

Biểu đồ triển khai mô tả kiến trúc hệ thống của phần cứng khác nhau như bộ xử lý, các thiết bị và các thành phần phần mềm thực hiện trên kiến trúc đó.



Hình 3.42 Biểu đồ triển khai hệ thống

Hệ thống Quản lý thẻ tại Bưu điện Thành phố Hà Nội khi triển khai đầy đủ sẽ được xây dựng theo biểu đồ triển khai trong hình 3.41, trong đó:

- Máy Dịch vụ CSDL: Là máy chứa CSDL của hệ thống quản lý thẻ.

- Máy May1, May2: là các máy trạm tại Bưu điện Tp.HN thực hiện chương trình quản lý thẻ.

- Máy Dịch vụ Datmua: Chứa trình dịch vụ Internet phục vụ việc đặt mua thẻ qua mạng.

- ISP: Nhà cung cấp dịch vụ Internet.

- Máy Đại lý: Là máy tính của đại lý có kết nối Internet để có thể đặt mua thẻ qua Internet.

Tóm lược: Chương 3 đã trình bày kết quả ứng dụng phương pháp hướng đối tượng, quy trình RUP và mẫu thiết kế vào việc phân tích, thiết kế, xây dựng Hệ thống quản lý thẻ điện thoại trả trước tại Bưu điện Thành phố Hà Nội. Đặc biệt việc ứng dụng các mẫu thiết kế đã hỗ trợ rất nhiều cho việc quyết định gán trách nhiệm cho từng lớp thiết kế, xây dựng cấu trúc cũng như mối quan hệ giữa các lớp để làm cho hệ thống có khả năng tái sử dụng cao.

ii) Biểu đồ trình tự xuất thẻ	87
iii) Biểu đồ trình tự thanh toán bằng tiền mặt	88
3.4.2 giao kèo thao tác cỦa hỆ thỔng	88
3.4.2.1 Hợp đồng nhập thẻ	89
3.4.2.2 Hợp đồng kết thúc phiên nhập thẻ	89
3.4.2.3 Hợp đồng xuất thẻ	90
3.4.2.4 Hợp đồng kết thúc phiên xuất thẻ	90
3.4.2.5 Hợp đồng tính dư nợ	91
3.4.2.6 Hợp đồng trả bằng tiền mặt	91
3.4.2.7 Hợp đồng khởi động hệ thống	92
3.5 ThiẾT kẾ hỆ thỔng	92
3.5.1 Các biỂu đồ cỘng tác	92
3.5.1.1 Biểu đồ cộng tác cho Hợp đồng nhập thẻ	92
3.5.1.2 Biểu đồ cộng tác cho Hợp đồng kết thúc phiên nhập thẻ ...	93
3.5.1.3 Biểu đồ cộng tác cho Hợp đồng xuất thẻ	94
3.5.1.4 Biểu đồ cộng tác cho Hợp đồng kết thúc phiên xuất thẻ	94
3.5.1.5 Biểu đồ cộng tác cho Hợp đồng tính dư nợ của đại lý	95
3.5.1.6 Biểu đồ cộng tác cho Hợp đồng thanh toán bằng tiền mặt .	97
3.5.1.7 Biểu đồ cộng tác cho Hợp đồng khởi động hệ thống	98
3.5.2 BiỂu đồ lỚp THIẾT KẾ	99
3.5.2.1 Thiết kế lớp	99
3.5.2.2 Bổ sung quan hệ giữa các lớp	100
3.5.2.3 Bổ sung quan hệ phụ thuộc	101

Hình:

Bảng 3.1 Các chức năng cơ bản của Hệ thống quản lý thẻ.....	70
Hình 3.3 Biểu đồ UC của hệ thống	77
Hình 3.4 Giao diện cho việc nhập thẻ.....	78
Hình 3.5 Giao diện cho việc xuất thẻ.....	79
Hình 3.6 Giao diện cho việc thanh toán.....	80
Hình 3.10 Biểu đồ trình tự nhập thẻ.....	87
Hình 3.11 Biểu đồ trình tự xuất thẻ.....	87
Hình 3.12 Biểu đồ trình tự thanh toán bằng tiền mặt.....	88
Hình 3.13 Biểu đồ cộng tác hợp đồng nhập thẻ.....	93
Hình 3.14 Biểu đồ cộng tác hợp đồng kết thúc phiên nhập thẻ	93
Hình 3.15 Biểu đồ cộng tác xuất thẻ	94
Hình 3.16 Biểu đồ cộng tác hợp đồng kết thúc phiên xuất thẻ	95
Hình 3.17 Biểu đồ cộng tác khởi tạo dư nợ	95
Hình 3.18 Biểu đồ cộng tác tính dư nợ	96
Hình 3.19 Biểu đồ cộng tác tính dư nợ	97
Hình 3.20 Biểu đồ cộng tác khởi động hệ thống	98
Hình 3.27 Gói Domain Concepts	104
Hình 3.33 Biểu đồ trình tự thanh toán bằng thẻ tín dụng	107
Hình 3.34 Biểu đồ trình tự thanh toán bằng séc.....	107
Hình 3.35 Polymorphism trong xác thực thanh toán	110

Hình 3.36 Biểu đồ cộng tác khởi tạo hợp đồng TrabangThe	111
Hình 3.37 Biểu đồ cộng tác khởi tạo hợp đồng TrabangSec	111
Hình 3.38 Áp dụng Mẫu Singleton thiết kế lớp XuatThe và Thanhtoan.....	112
Hình 3.39 Áp dụng mẫu Remote proxy: (a) tìm dịch vụ xác thực, (b) sử dụng Remote proxy và (c) các hành động Remote proxy.....	113
Hình 3.40 Biểu đồ quan hệ dữ liệu của hệ thống.....	114
Hình 3.41 Biểu đồ thành phần Hệ thống quản lý thẻ.....	115
Hình 3.42 Biểu đồ triển khai hệ thống	116

PHẦN KẾT LUẬN

Luận văn đã trình bày những nghiên cứu về phương pháp phân tích, thiết kế hướng đối tượng, tiến trình phát triển phần mềm RUP (*Rational Unified Process*) và mẫu thiết kế. Trong đó, phần mẫu thiết kế tập trung nghiên cứu và trình bày về các mẫu GRASP (mẫu của những nguyên tắc chung trong ấn định trách nhiệm) và GoF (*Gang of Four*).

Các kết quả nghiên cứu đã được ứng dụng vào việc phân tích, thiết kế, xây dựng thử nghiệm Hệ thống quản lý thẻ điện thoại trả trước tại Bưu điện Thành phố Hà Nội. Hệ thống được xây dựng theo phương pháp lập trình hướng đối tượng. Việc phân tích, thiết kế được thể hiện bằng ngôn ngữ UML thông qua công cụ Rational Rose và thực hiện theo quy trình RUP. Việc áp dụng một số mẫu thiết kế GRASP và GoF đã làm cho phân tích, thiết kế được thuận lợi và hiệu quả hơn, giúp cho chương trình có khả năng tái sử dụng cao hơn.

Tuy nhiên với thời gian có hạn và nhiều kiến thức còn mới nên luận văn chắc chắn còn nhiều hạn chế. Chính vì vậy trong thời gian tới em mong muốn được tiếp tục nghiên cứu, tìm hiểu sâu hơn về các mẫu thiết kế và áp dụng thực hiện quy trình RUP cho việc xây dựng các bài toán lớn tại cơ quan.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Nguyễn Văn Ba (2005), *Phân tích và thiết kế hệ thống thông tin*, NXB Đại học quốc gia Hà Nội.
2. Đặng Văn Đức (2002), *Phân tích thiết kế hướng đối tượng bằng UML*, NXB Giáo dục.
3. Nguyễn Quý Minh, Tăng Nguyễn Trung Hiếu, Phạm Anh Vũ, Lê Hải Dương, Phương Lan (2005), *Design patterns*, NXB Phương đông .

Tiếng Anh

4. C. Larman (1998), *Applying UML and Patterns*, Prentice Hall PTR .
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides (1994), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.
6. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal (1996), *Pattern-Oriented Software Architecture - A System of Patterns (Vol.1)*, Wiley and Sons.
7. G. Booch, J. Rumbaugh, I. Jacobson (1999), *The Unified Modeling Language User Guide*, Addison Wesley.
8. P. Kruchten (2000), *The Rational Unified Process: An Introduction (2nd Edition)*, Addison-Wesley Professional.
9. W. Pree (1995), *Design Patterns for Object-Oriented Software Development*, Addison-Wesley.
10. Z. Liu (2002), *Object-Oriented Software Development with UML*, UNU/IIST Report No.259.