Phân tích thiết kế hệ thống với UML

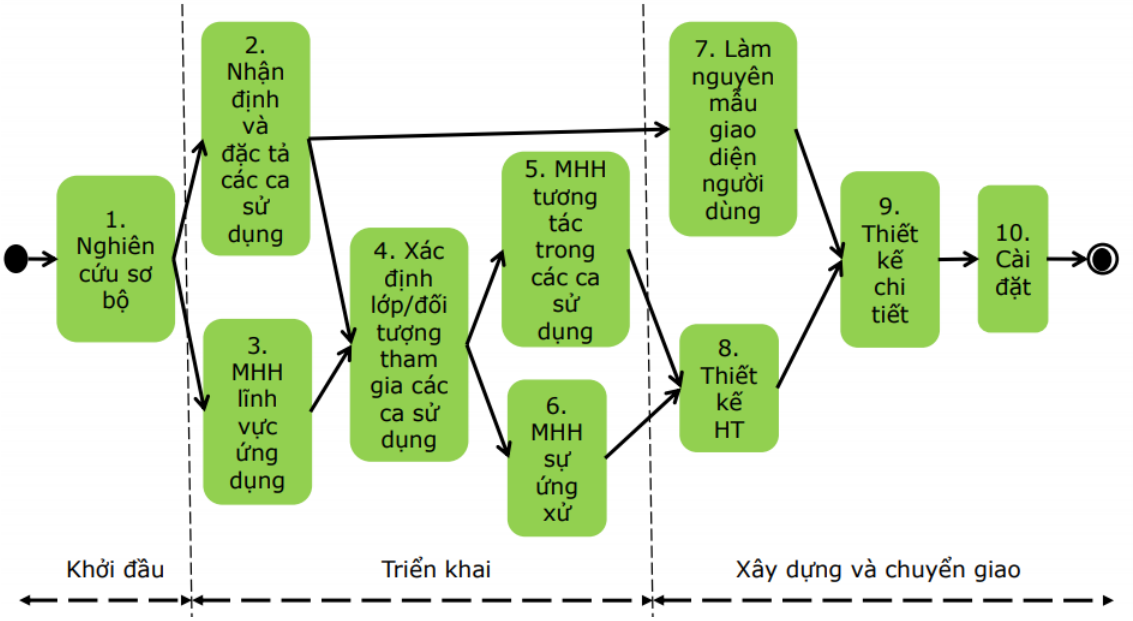
Ngô Quang Dương

Ngày 29 tháng 3 năm 2019

Tóm tắt nội dung

Trong tài liệu này, mình trình bày một cách ngắn gọn những điều cần biết về việc phân tích thiết kế hệ thống với **UML** - các loại sơ đồ **UML**, các giai đoạn trong phân tích thiết kế hệ thống và những việc được thực hiện trong những giai đoạn đó.

Mục lục



1 Các loại sơ đồ

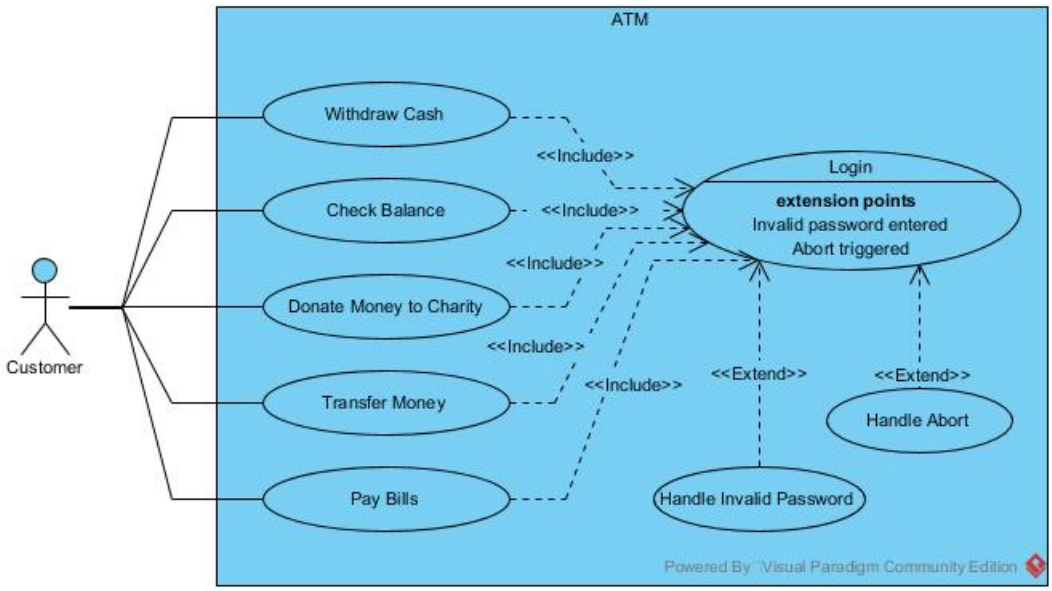
1.1 Sơ đồ ca sử dụng

Thường được gọi nhiều hơn bằng cái tên *sơ đồ use case*, sơ đồ này thể hiện các *tác nhân* hệ thống (là những thành phần có tương tác với hệ thống nhưng không nằm trong hệ thống) và các *chức năng* đối với từng tác nhân.

Trong một *sơ đồ use case* thông thường, các *tác nhân* được biểu diễn bằng hình người que, còn các *chức năng*(use case) được biểu diễn bằng hình elip với nội dung là tên của *chức năng* đó. Hãy xem qua ví dụ sau:

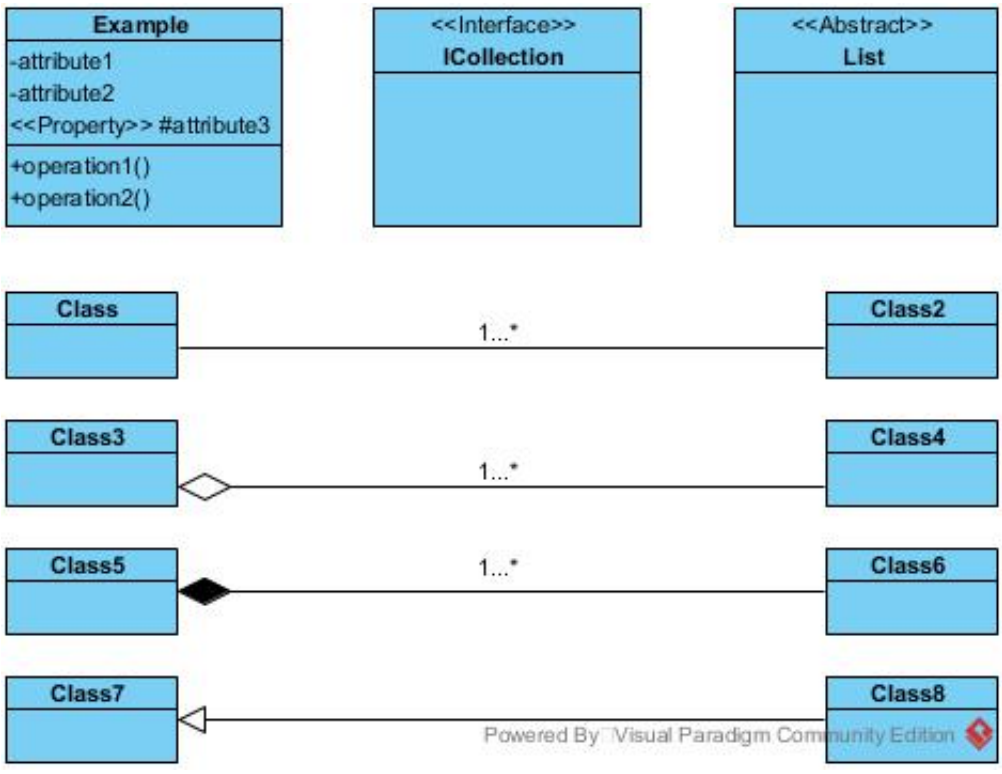
Giải thích các ký hiệu:

- *Khung hình chữ nhật*: đại diện cho hệ thống, cạnh của hình chữ nhật là biên hệ thống, những gì nằm trong là thành phần hệ thống, còn nằm ngoài thì không.
- *Người que*: tác nhân.
- *Hình elip*: use case.
- *Đường nét liền nối người que với hình elip*: thể hiện rằng tác nhân có thể sử dụng use case đó.
- *Mũi tên nét đứt với label «include»*: thể hiện rằng use case này(gốc mũi tên) chứa use case kia(ngọn mũi tên).
- *Hình elip có extension point*: use case có điểm mở rộng - tức là khi use case đó được thực hiện thì có thể có những tình huống khác xảy ra - như trong sơ đồ trên thì việc đăng nhập có thể đúng hoặc sai, mỗi tình huống đó là một extension point.
- *Mũi tên nét đứt với label «extend»*: thể hiện rằng use case này(gốc mũi tên) sẽ mở rộng use case kia(ngọn mũi tên), với điều kiện được trình bày trong extension point.



Hình 1: Sơ đồ use case cho hệ thống ATM

1.2 Sơ đồ lớp



Hình 2: Minh họa các ký hiệu trong sơ đồ lớp

Sơ đồ lớp (*class diagram*) là sơ đồ thể hiện nội dung của các lớp (thuộc tính, phương thức, bổ từ truy cập) và mối quan hệ giữa các lớp (kế thừa, kết tập, kết hợp, hợp thành).

Giải thích các ký hiệu:

- *Khung hình chữ nhật*:
  - *Khoang trên cùng*: tên class, phần «...» trên cùng là stereotype – để thể hiện những kiểu mà UML không có sẵn như là interface, abstract, ... nếu không có gì thì mặc định hiểu đó là class.
  - *Khoang thứ hai*: danh sách các thuộc tính (có thể bổ sung kiểu dữ liệu).
  - *Khoang thứ ba*: danh sách các phương thức (có thể bổ sung kiểu dữ liệu trả về và các tham số cùng kiểu dữ liệu tương ứng).
  - *Dấu -*: `private`.
  - *Dấu #*: `protected`.
  - *Dấu +*: `public`.
- *Đường nét liền*: thể hiện quan hệ kết hợp - tức là hai class có tương tác với nhau.
- *Đường nét liền có một đầu hình thoi trắng*: thể hiện quan hệ kết tập – class này(có hình thoi trắng) sở hữu một object của class kia(không có hình thoi trắng) nhưng object của class kia có thể tồn tại độc lập với class này.
- *Đường nét liền có một đầu hình thoi đen*: thể hiện quan hệ hợp thành – cũng tương tự như quan hệ kết tập nhưng khác là object của class kia chỉ tồn tại nếu có object của class này.
- *Bội số(1...*, *, 1*: trong mối quan hệ kết hợp với bộ 1 thì object của class bên trái sẽ kết hợp với 1 object của class bên phải; 1...* thì object của class bên trái kết hợp với ít nhất 1 object của class bên phải; * thì object bên trái có thể kết hợp với không hoặc nhiều object của class bên phải.
- *Mũi tên có ngọn hình tam giác trắng*: thể hiện quan hệ tổng quát hóa – class ở ngọn mũi tên tổng quát class ở gốc mũi tên.

1.3 Sơ đồ tuần tự

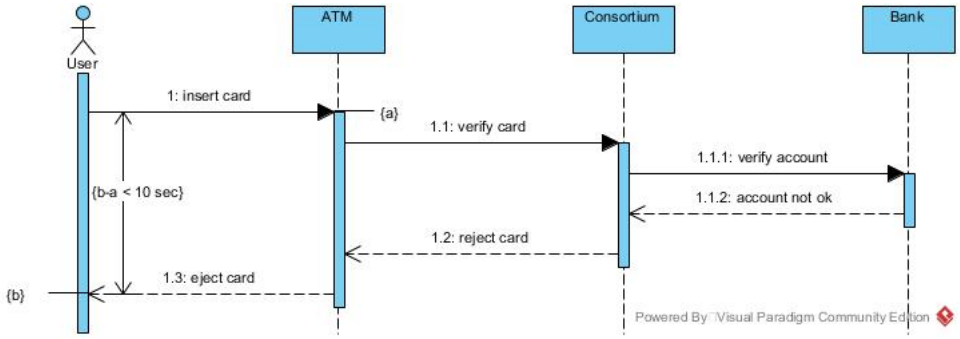
Sơ đồ tuần tự (sequence diagram) được sử dụng để mô tả các thao tác được thực hiện trong một use case – cụ thể là các object tương tác với nhau như thế nào, gọi tới những phương thức nào.

Các bước của quá trình tương tác rất tiên theo dõi vì chúng được đặt theo thứ tự từ trái sang phải, từ trên xuống dưới và còn được đánh số, sử dụng các mũi tên.

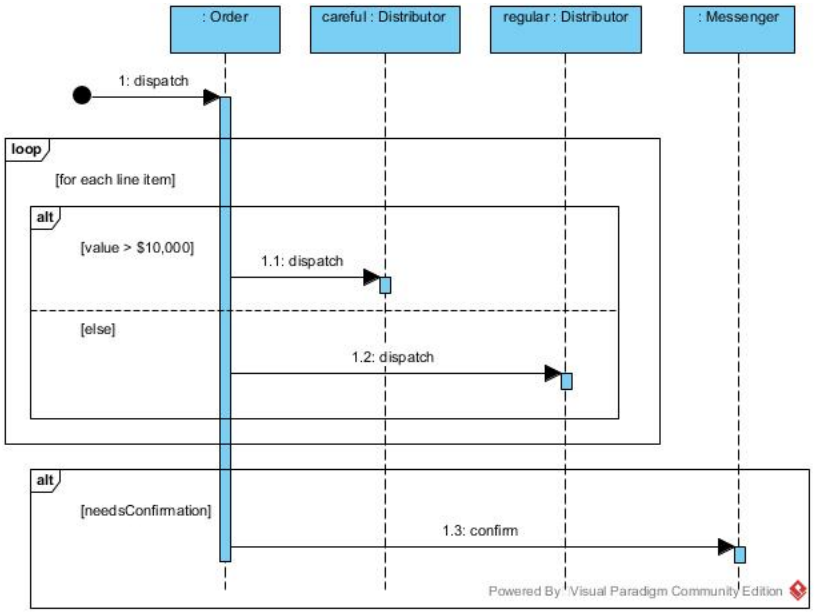
Hãy quan sát sơ đồ tuần tự thể hiện quá trình cho thẻ vào máy ATM :

Giải thích các ký hiệu:

- *Hình người que*: tác nhân.
- *Hình chữ nhật có nhãn*: object.
- *Đường nét đứt thẳng*: đường sống (life line).
- *Hình chữ nhật dài trên đường sống*: thời gian hoạt động của object hoặc tác nhân đó.
- *Mũi tên nét liền*: gọi phương thức của object được trỏ đến.
- *Mũi tên nét đứt*: kết quả trả về.



Hình 3: Sơ đồ tuần tự minh họa việc cho thẻ vào máy ATM



Hình 4: Ví dụ về cấu trúc lặp và rẽ nhánh trong sơ đồ tuần tự

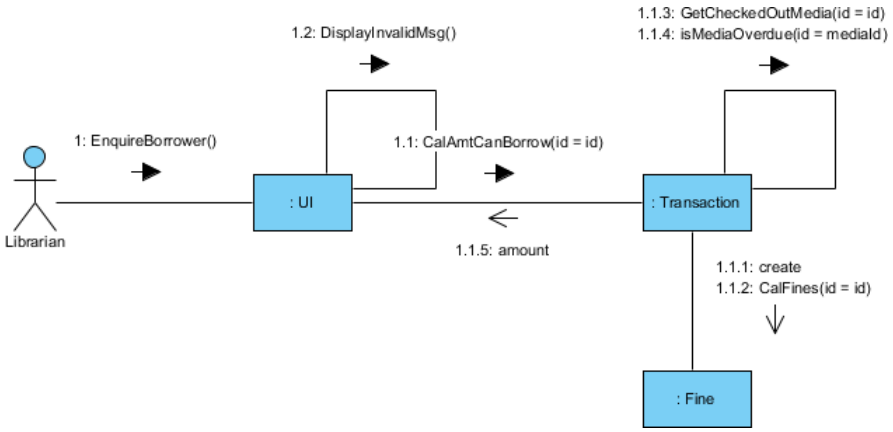
Có thể thấy rằng thứ tự được gọi của các phương thức được thể hiện qua cách đánh số và cách sắp xếp từ trái sang phải, từ trên xuống dưới. Ngoài ra, trong sơ đồ tuần tự, người ta còn có thể trình bày các cấu trúc lặp, rẽ nhánh, `continue`, `break`. Trong đó, cấu trúc lặp, rẽ nhánh sẽ được đặt vào một khung có nhãn lần lượt là **loop** và **alt**.

1.4 Sơ đồ giao tiếp

Sơ đồ giao tiếp(communication diagram) có tác dụng giống hệt như sơ đồ tuần tự. Ưu điểm của nó so với sơ đồ tuần tự là sơ đồ giao tiếp có thể trình bày tự do hơn, trong khi sơ đồ tuần tự có một cấu trúc cố định (sẽ càng chiếm nhiều diện tích hơn khi số lượng object hoặc các bước tăng lên). Nhược điểm là nó không tiện theo dõi thứ tự các bước như trong sơ đồ tuần tự.

1.5 Sơ đồ hoạt động

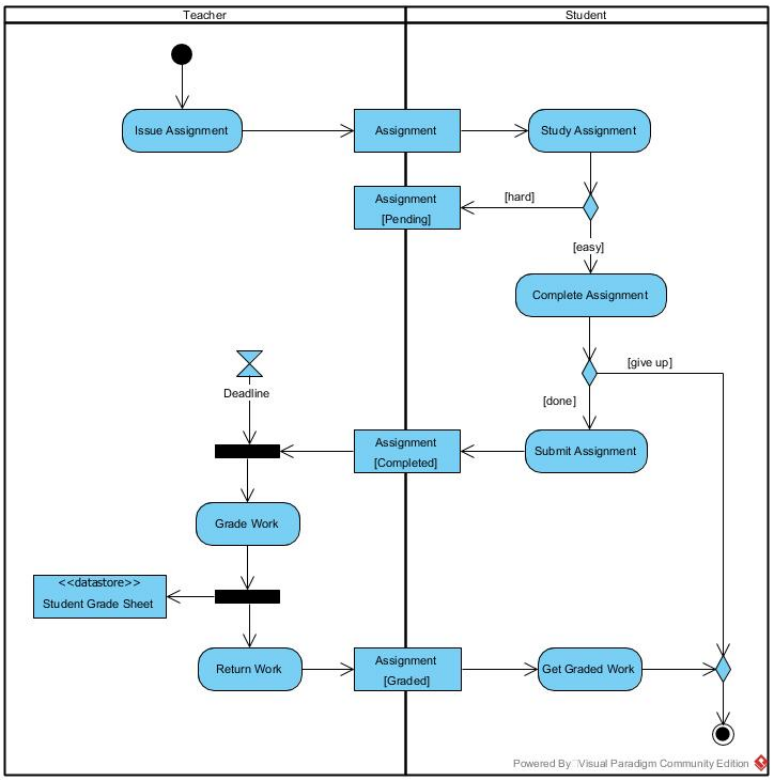
Sơ đồ hoạt động(activity diagram) khá giống với sơ đồ luồng, sơ đồ khối (flowchart). Nếu như sơ đồ tuần tự và sơ đồ giao tiếp thể hiện các object và lời gọi các phương thức thì ở sơ đồ hoạt động, quá



Hình 5: Sơ đồ giao tiếp thể hiện quá trình mượn sách

trình đó được thể hiện chi tiết hơn – có thể hiểu là sơ đồ này sẽ trình bày cả những điều được thực hiện bên trong phương thức.

Sơ đồ hoạt động luôn bắt đầu bằng một node bắt đầu (một hình tròn đen) và đích đến là node kết thúc (hình tròn đen có viền). Một sơ đồ hoạt động luôn có đúng một node bắt đầu nhưng có thể có nhiều node kết thúc (tương ứng với nhiều tình huống khác nhau).

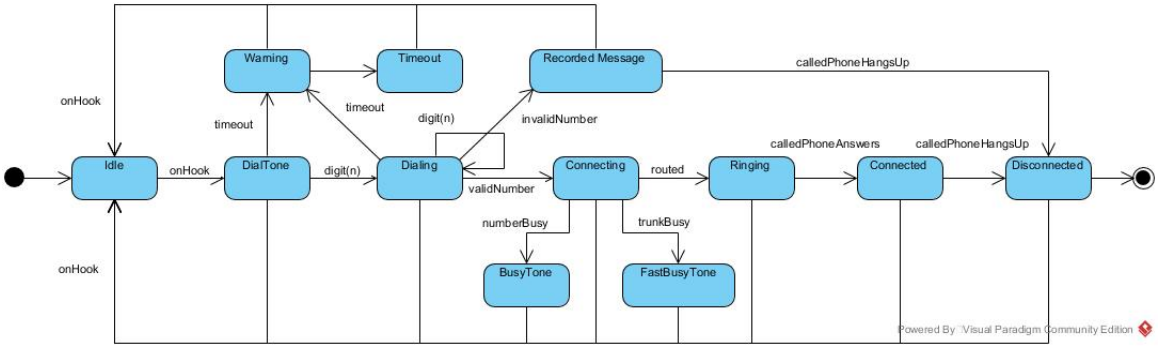


Hình 6: Sơ đồ hoạt động

Các bước của một quá trình được thể hiện trong sơ đồ hoạt động bằng các hình chữ nhật có bo tròn góc. Các mũi tên sẽ trỏ luồng hoạt động. Cũng giống như trong sơ đồ khối, sơ đồ hoạt động cũng có thể thể hiện cấu trúc rẽ nhánh (bằng node quyết định hình thoi) và cấu trúc lặp.

1.6 Sơ đồ máy trạng thái

Trong lập trình hướng đối tượng, một đối tượng có hai đặc tính: thứ nhất là thuộc tính(property or attribute), thứ hai là phương thức(method or operation). Trong đó, thuộc tính cũng chính là *trạng thái* của đối tượng.



Hình 7: Minh họa sơ đồ máy trạng thái cho hệ thống điện thoại

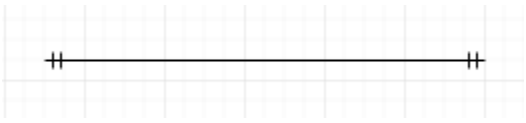
Sơ đồ máy trạng thái(state machine diagram) chỉ ra trạng thái của hệ thống và những sự kiện, phương thức có thể kích hoạt sự thay đổi của những thuộc tính đó.

1.7 Sơ đồ quan hệ thực thể

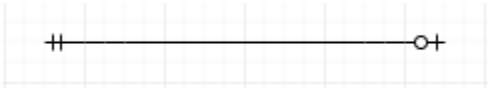
Sơ đồ quan hệ thực thể(entity relationship diagram) thể hiện mối quan hệ giữa các thực thể được lưu trữ trong một cơ sở dữ liệu. Thực thể trong ngữ cảnh này chính là những object. Một tập thực thể là một bộ các thực thể cùng loại. Các thực thể có thể có một hoặc nhiều thuộc tính.

Mối quan hệ trong sơ đồ quan hệ thực thể có thể thuộc một trong các loại sau, còn được gọi là những bội số. Bội số nếu có thể nhận giá trị 0 được gọi là *optional*(tùy chọn), ngược lại được gọi là *mandatory*(bắt buộc).

- **1 – 1**. Quan hệ một – một, được thể hiện bằng đường nối:

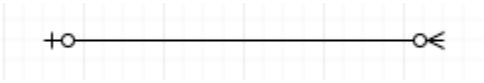


Hình 8: 1 – 1

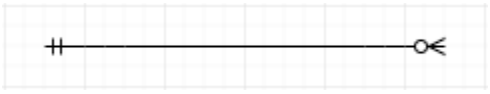


Hình 9: 1 – 0...1

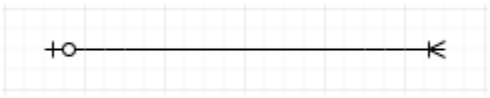
- **1 – n**. Quan hệ một – nhiều:



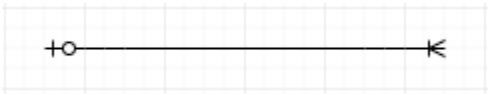
Hình 10: 0..1 – 0..n



Hình 11: 1 – 0..n

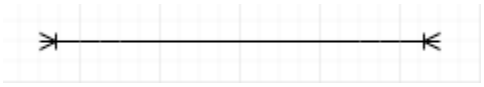


Hình 12: 0..1 – n

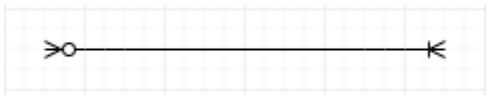


Hình 13: 1 – n

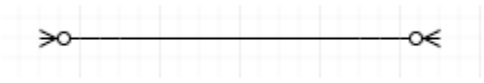
- **n – n**. Quan hệ nhiều – nhiều:



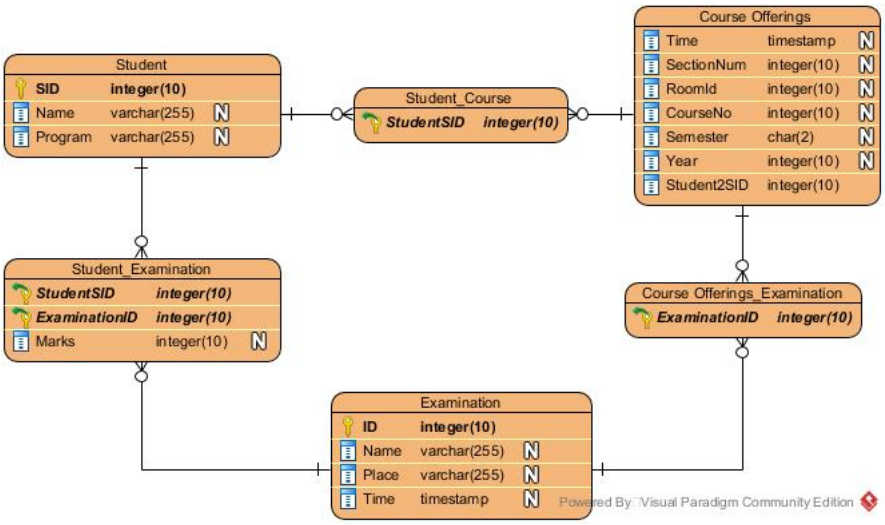
Hình 14: n – n



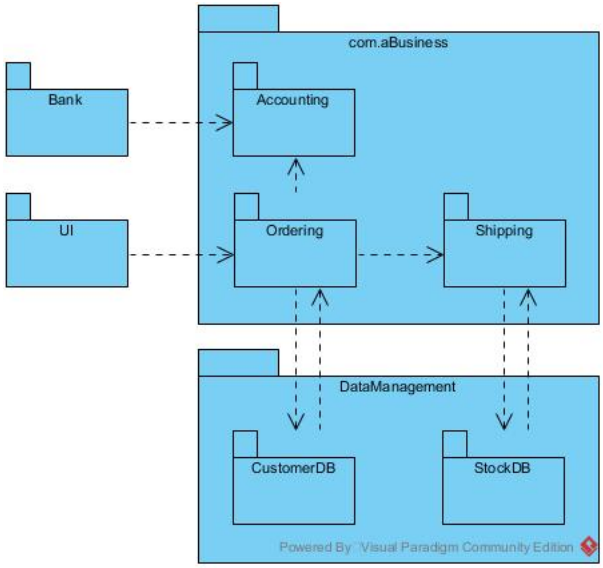
Hình 15: 0..n – n



Hình 16: 0..n – 0..n



Hình 17: Ví dụ sơ đồ quan hệ thực thể cho cơ sở dữ liệu lưu trữ điểm của sinh viên



Hình 18: Sơ đồ gói

1.8 Sơ đồ gói

Một tính chất khác của lập trình hướng đối tượng là tính đóng gói. Có hai khía cạnh của tính đóng gói: thứ nhất là sự giới hạn truy cập một thuộc tính, phương thức của đối tượng; thứ hai là sự nhóm họp của các class có cùng chức năng. Khía cạnh thứ nhất đã được thể hiện trong *sơ đồ lớp*, còn khía cạnh thứ hai được thể hiện trong sơ đồ gói(package diagram)

Giải thích các ký hiệu:

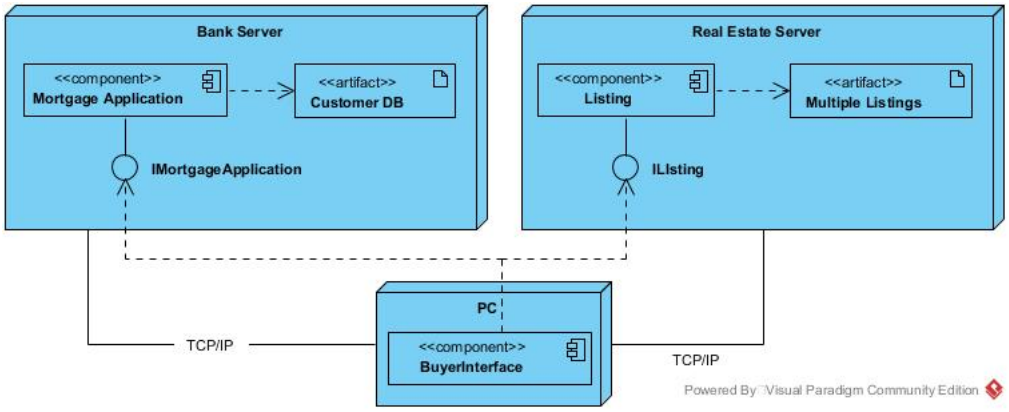
- *Hình tập tài liệu*: đại diện cho một package, các package có thể rời nhau hoặc package này chứa một hay nhiều package con.

- *Mũi tên nét đứt*: quan hệ phụ thuộc, package ở gốc mũi tên phụ thuộc vào package ở ngọn mũi tên.

Ngoài ra trong các package trên sơ đồ gói có thể biểu diễn thêm các class, interface bên trong.

1.9 Sơ đồ triển khai

Sơ đồ triển khai (deployment diagram) được sử dụng để mô tả sự bố trí các *thành phần thực thi*.



Hình 19: Sơ đồ triển khai

Các thành phần đó có thể là phần cứng(nút cứng, các thiết bị vật lý, ...) hoặc phần mềm(nút mềm, các file thực thi, server, ...). Các nút trong một sơ đồ triển khai được kết nối với nhau.

Giải thích các ký hiệu:

- *Hình hộp*: một nút.
- *Các stereotype*: các thành phần trong một nút.
- *Hình tròn*: giao diện(interface).
- *Mũi tên nét đứt*: thể hiện việc sử dụng, cài đặt giao diện.

2 Các giai đoạn

2.1 Phân tích yêu cầu

2.2 Đặc tả yêu cầu

2.3 Xác định các class/object phân tích

2.4 Mô tả quá trình tương tác trong một use case

2.5 Thiết kế hệ thống con

2.6 Lựa chọn công nghệ

2.7 Cài đặt và triển khai