**PHA LẤY YÊU CẦU**

**1: Các công việc cần thực hiện**

* Xác định những gì khách hàng cần chứ không phải những gì khách hàng muốn

Nội dung cần thực hiện:

* Xác định lĩnh vực của ứng dụng: Môi trường cụ thể mà sản phẩm phần mềm đích hoạt động
* Xây dựng danh sách các từ/ thuật ngữ/khái niệm chuyên môn
* Các từ/thuật ngữ/khái niệmcần được giải thích rõ ràng ý nghĩa của chúng
* Xây dựng mô hình nghiệp vụ: mô hình các tiền trình nghiệp vụ của khách hàng
* Sử dụng các kỹ thuật (phỏng vẫn là chính, ngoài ra còn dùng: dùng bản thăm dò ý kiến, quan sát trực tiếp,…) để có bản mô tả nghiệp vụ của khách hàng
* Từ bản mô tả đó, sử dụng UML để biểu diễn các yêu cầu cầu của khách hàng ( dưới dạng Use case)
* Xác định các yêu cầu của khách hàng
* Dựa trên mô hình nghiệp vụ xác định các yêu cầu ban đầu => tinh chỉnh làm mịn
* Các yêu cầu là động – thay đổi thường xuyên thay đổi: cách giải quyết: duy trì một danh sách các yêu cầu quan trọng cùng các use case của các yêu cầu đã được khách hàng phê duyệt
* Có 2 loại yêu cầu: yêu cầu chức năng và yêu cầu phi chức năng
* Lặp đi lặp lại các bước trên cho đến khi khách hàng đồng ý

**2: Các kỹ thuật áp dụng**

* Ký thuật phỏng vấn: gặp gỡ trực tiếp khách hàng và người dùng để thu thập thông tin
* Có 2 loại câu hỏi: cầu hỏi kết thúc đóng và câu hỏi kết thúc mở
* Có 2 kiểu phỏng vấn: phỏng vấn có cầu trúc và phỏng vấn không có cấu trúc
* Người phỏng vấn phải quen thuộc với lĩnh vực ứng dụng, luôn sẵn sàng tiếp thu cái mới
* Sau khi phỏng vấn xong, nên chuẩn bị một bản tường trình được viết ra đưa cho người được phỏng vấn
* Một số kỹ thuật khác:
* Một bản thăm dò ý kiến rất hữu ích khi lấy ý kiến của hàng trăm người.
* Kiểm tra các định dạng nghiệp vụ mà chỉ ra cách khách hàng thực hiện những công việc nghiệp vụ
* Quan sát trực tiếp người dùng:
* Máy quay là một phiên bản hiện đại của kỹ thuật này
* Nhược điểm: cần rất nhiều thời gian để phân tích các băng video, người được quay có thể xem máy quay vì sự xâm phạm tùy tiện đời sống riêng tư

**3:Các kết quả (artifact) cần đạt**

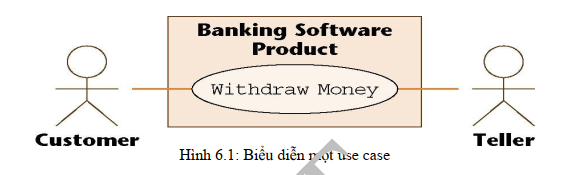
* Xác đinh đươc miền ứng dụng
* Tìm hiểu từ/thuật ngữ/khái niệm chuyên môn và ý nghĩa của chúng
* Các actor liên quan
* Biểu đồ use case

**4:Các ký pháp UML**

* Một use case mô hình hóa sự tương tác giữa bản thân sản phẩm phần mềm và những người sử dụng sản phẩm phần mềm đó (các actor) hoặc là cho thấy sự tương tác giữa sản phẩm phần mềm và môi trường mà sản phẩm phần mềm hoạt động. (một tác nhân là một thành viên của thế giới bên ngoài sản phẩm phần mềm.)
* Cách xác định actor:
* Một actor thường là người dùng sản phẩm phần mềm
* Người khởi xướng một use case hoặc một người nào đó đóng một vai trò quan trọng trong use case cũng có thể được coi là một actor, bất kể người đó có phải là người dùng sản phẩm phần mềm hay không.
* Một actor của hệ thống có thể đóng nhiều vai trò
* Actor không nhất thiết phải là người. trong nhiều trường hợp, một sản phẩm phần mềm khác có thể là một người dùng

Lưu ý: tránh việc xác định các actor chồng chéo

Ví dụ:



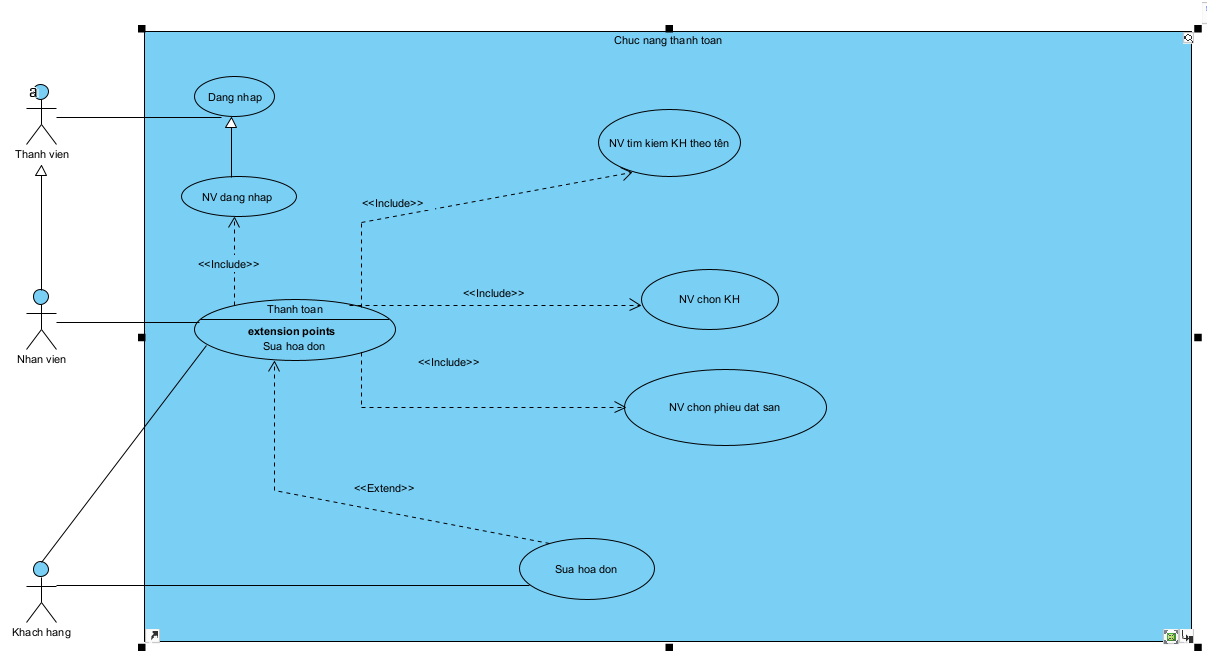
**5: Áp dụng cho case study Thanh toán của Hệ thống quản lí cho thuê sân bóng mini (13.5)**

Đề bài: Modul "***Khách hàng thanh toán***": Khi KH đến yêu cầu thanh toán, NV chọn menu tìm phiếu đặt sân theo tên KH → nhập tên KH+click tìm kiếm → hệ thống hiển thị danh sách các KH có tên vừa nhập → NV chọn tên KH đúng với thông tin KH hiện tại → hệ thống hiện lên danh sách các phiếu đặt mà KH đó đang đặt → NV click vào nút chọn thanh toán cho 1 phiếu đặt sân → hệ thống hiện hóa đơn đầy đủ thông tin khách hàng + 1 bảng danh sách các sản phẩm ăn uống mà KH đã sử dụng trong suốt các buổi thuê sân như mô tả trên + dòng cuối là tổng số tiền trả → NV click xác nhận → hệ thống cập nhật vào CSDL.

* Miền ứng dụng (domain): lĩnh vực bóng đá (quản lý cho thuê sân bóng mini)
* Danh sách các từ chuyên môn trong lĩnh vực :

|  |  |
| --- | --- |
| Phiếu đặt sân | Là hợp đồng cho thuê sân khi khách hàng đến yêu cầu thuê sân |
| Sân | Là một loại hình thi đấu của môn bóng đá, thường có hình chữ nhật |
| Hoá đơn | Là chi tiết các đề mục mà người thuê sân cần phải trả |
| Buổi thuê sân | Là các buổi trong tuần thuê sân |
| Sản phẩm ăn uống | Là các sản phẩm mà khách hàng sử dụng trong quá trình thuê sân |
| Tổng tiền | Là tổng số tiền mà người thuê phải trả. |
| Tiền cọc | Là 1 phần tiền cần trả trước khi đặt thuê sân |

* Các actor:
* Nhân viên: người trực tiếp làm việc với hệ thống. Chức năng: Thanh toán.
* Khách hàng: Người không trực tiếp làm việc với hệ thống, nhưng có vai trò quan trọng với phần mềm.
* Mục đích của case study: giúp phục vụ công tác quản lý của nhân viên, ở đây cụ thể là modul Thanh toán
* Biểu đồ use case chi tiết modul thanh toán:



**PHA PHÂN TÍCH**

**1: Các công việc cần thực hiện**

* Mục đích của pha phân tích là để hiểu sâu hơn về các yêu cầu, là mô tả những yêu cầu đó theo cách mà thiết kế và triển khai ra kết quả được
* Các bước cần thực hiện:
* Trích xuất các lớp thực thể, lớp biên và lớp điều khiển
* Xác định mối quan hệ tương tác giữa các lớp
* Trích xuất các lớp thực thể: gồm ba bước được thực hiện lặp đi lặp lại và tăng dần:
* Bước 1: Mô hình hoá chức năng: Trình bày các kịch bản chuẩn (scenario)
* Bước 2: Mô hình hoá các lớp: Xác định các lớp thực thể, các thuộc tính, mối quan hệ tương tác => biểu đồ lớp thực thể (class diagram). Các cách thực hiện:
* Sử dụng kỹ thuật Trích xuất các danh từ (nêu ở phần kỹ thuật)
* Thẻ CRC mô hình các quan hệ giữa các lớp: C: class; R: Responsibility; C: collaboration
* Suy luận ra từ các lớp từ các use case và kịch bản của các use case
* Bước 3: Mô hình hoá hoạt động: Xác định các hoạt động được thực hiện bởi mỗi lớp thực thể hoặc lớp con => biểu đồ trạng thái (statechart)
* Các sự kiện, hoạt động được trích ra từ scenario
* Khi có một even xảy ra, trạng thái hiện tại sẽ được chuyển sang một trạng thái khác được biểu diễn bằng một mũi tên, trên mũi tên đó là tên sự kiện
* Trích xuất các lớp biên:
* Mỗi màn hình đầu vào, màn hình đầu ra và báo cáo in => đề xuất làm một lớp biên
* Các thông báo cũng có thể đề xuất làm lớp biên
* Trích xuất lớp điều khiển:
* Nếu hệ thống nhỏ, một lớp điều khiển dùng chung cho cả hệ thông
* Nếu hệ thống phức tạp, đề xuất lớp điều khiển theo từng modul chức năng
* Scenario cuối pha phân tích => biểu đồ tuần tự

**2: Các kỹ thuật áp dụng**

* Kỹ thuật trích các danh từ để xác định lớp thực thể
* Mô tả nghiệp vụ, luồng của hệ thống đó
* Tìm các danh từ trong đó
* Với mỗi danh từ đó, có thể đề xuất làm một lớp thực thể hoặc làm thuộc tính trong lớp thực thể

**3: Các kết quả (artifact) cần đạt**

* Các kịch bản (Scenario)
* Trích xuất ra được các lớp thực thể, lớp biên, lớp điều khiển
* Biều đồ lớp thực thể
* Thẻ CRC
* Biểu đồ trạng thái
* Kịch bản cuối pha phân tích (Scenario cuối pha phân tích)
* Scenario cuối pha phân tích
* Biểu đồ tuần tự pha phân tích

**4: Các ký pháp UML**

* Có 3 loại lớp: Lớp thực thể, lớp biên, lớp điều khiển
* Lớp biên (view)
* mô hình hóa sự tương tác giữa sản phẩm phần mềm và các tác nhân của nó
* thường được liên kết với đầu vào và đầu ra
* Lớp điều khiển (control)
* Mô hình hóa các phép tính và thuật toán phức tạp
* Một hệ thống có thể có một hoặc nhiều lớp điều khiển
* Lớp thực thể (model)
* các lớp mô hình hóa thông tin tồn tại lâu dài, được coi là đầu tiên
* có các thuộc tính và các phương thức cơ bản như get và set
* UML của 3 lớp:



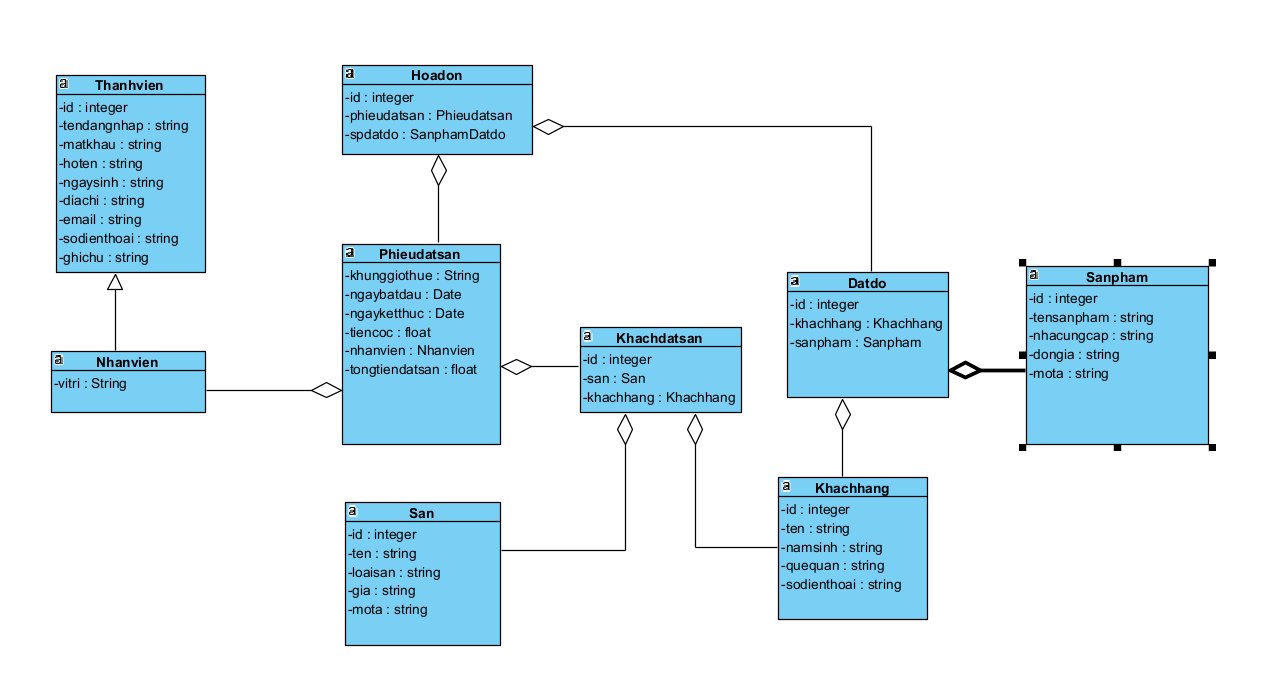
* Khi biểu diễn biểu đồ trạng thái bằng UML gồm các 3 phần chính: trạng thái, sự kiện và vị từ được phân bố trên toàn biểu đồ trang thái.

**5: Áp dụng cho case study Thanh toán:**

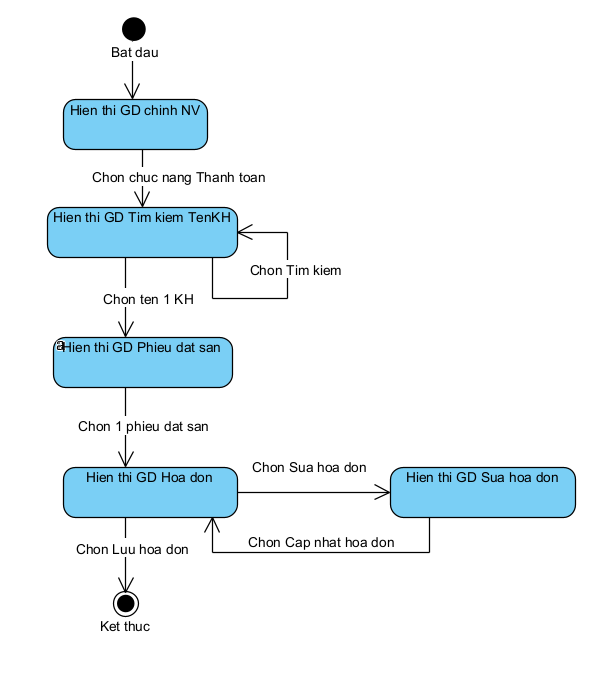
* Các kịch bản (Scenario)

|  |  |
| --- | --- |
| Kịch bản chính | 1. Sau khi đăng nhập, nhân viên chọn chức năng thanh toán từ giao diện màn hình chính. 2. Giao diện thanh toán hiện lên với menu tìm kiếm đặt sân theo tên KH và nút Tìm kiếm 3. Nhân viên nhập tên KH: Thắng + click tìm kiếm. 4. Giao diện danh sách KH Thắng hiện lên với các thông tin: tên KH, năm sinh, quê quán, ngày làm hợp đồng. 5. Nhân viên chọn đúng với thông tin khách hàng Thắng hiện tại. 6. Giao diện danh sách các phiếu mà Thắng đặt hiện lên với các thông tin: mã sân,tên khách hàng, ngày làm hợp đồng, khung giờ thuê trong tuần, ngày bắt đầu, ngày kết thúc. 7. Nhân viên chọn thanh toán cho một phiếu đặt sân 8. Giao diện hoá đơn hiện lên với đầy đủ:  * Dòng đầu có ngày làm hợp đồng, tên chủ sân, địa chỉ sân. * Thông tin khách hàng gồm có: tên khách hàng, năm sinh, quê quán, số điện thoại, khung giờ thuê trong tuần, ngày bắt đầu, ngày kết thúc,giá thuê một buổi, số tiền đã đặt cọc, ngày đặt cọc. * Bảng danh sách các sản phẩm ăn uống mà KH đã sử dụng gồm có các cột: mã sản phẩm, tên sản phẩm, nhà cung cấp(NCC), đơn giá,số lượng, thành tiền * Dòng cuối là tổng số tiền phải trả sau  1. Khi khách hàng có khiếu nại, nhân viên click vào sửa hoá đơn. 2. Giao diện sửa hoá đơn hiện lên với đầy đủ thông tin:  * Dòng đầu có ngày làm hợp đồng, tên chủ sân, địa chỉ sân. * Thông tin khách hàng gồm có: tên khách hàng, năm sinh, quê quán, số điện thoại, khung giờ thuê trong tuần, ngày bắt đầu, ngày kết thúc,giá thuê một buổi, số tiền đã đặt cọc, ngày đặt cọc. * Bảng danh sách các sản phẩm ăn uống mà KH đã sử dụng gồm có các cột: mã sản phẩm, tên sản phẩm, nhà cung cấp(NCC), đơn giá,số lượng, thành tiền * Dòng cuối là tổng số tiền phải trả sau * 2 nút Lưu và Huỷ  1. Nhân viên click vào lưu sau khi sửa xong các khiếu nại của khách hàng. 2. Nhân viên click vào huỷ để quay lại giao diện xem hoá đơn. 3. Khi khách hàng không có khiếu nại gì, nhân viên click vào lưu hoá đơn để xác nhận thanh toán và lưu vào CSDL. 4. Hệ thống thông báo thanh toán thành công và quay về giao diện chính. |
| Ngoại lệ | 3. Khách hàng không tồn tại  4. Nhập tên khách hàng xong, không có khách hàng nào hiện lên |

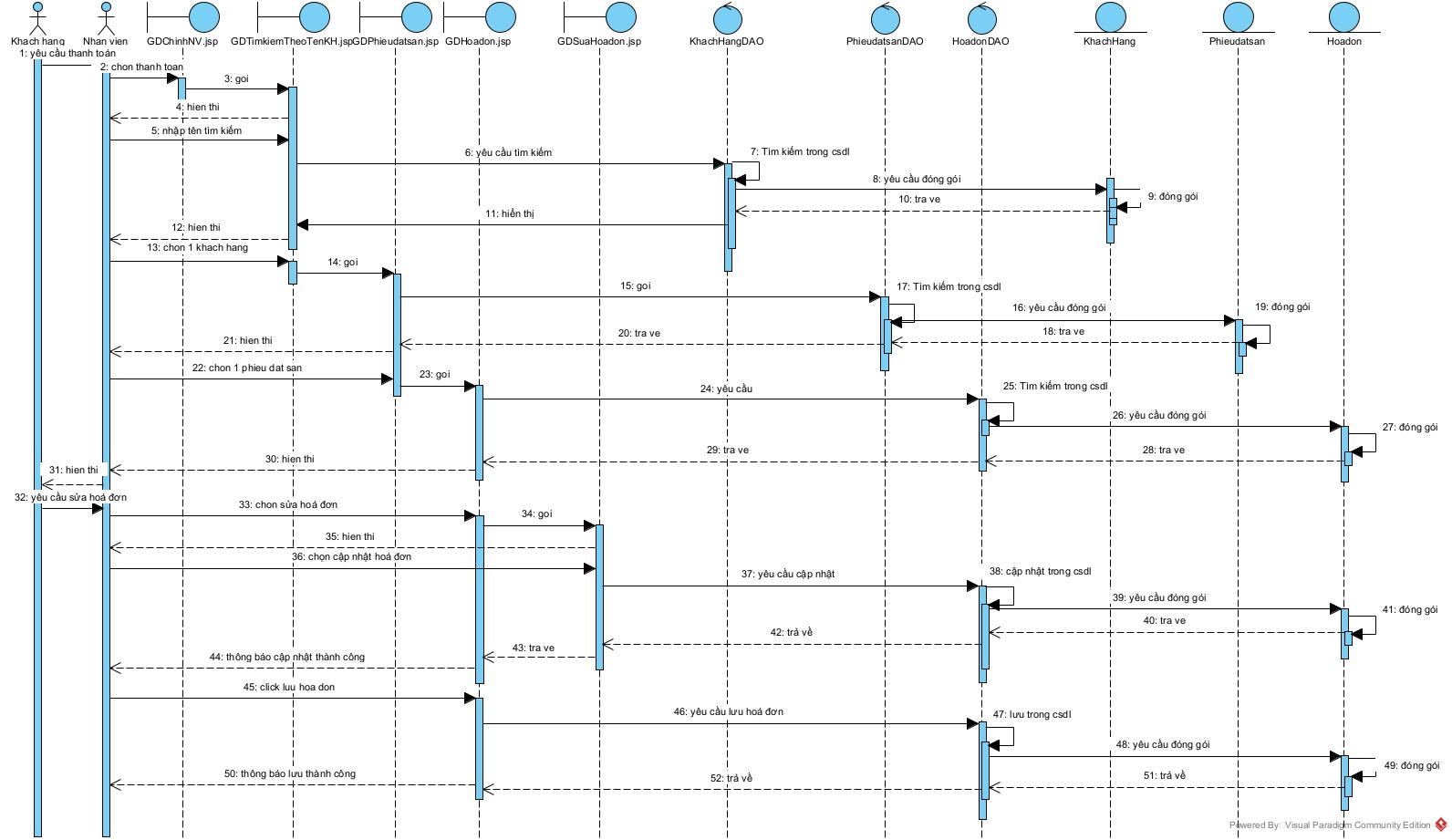
* Biều đồ lớp thực thể:



* Biểu đồ trạng thái

****

* Biểu đồ tuần tự:



**PHA THIẾT KẾ**

**1: Các công việc cần thực hiện**

* Đầu vào: tài liệu đặc tả, một bản mô tả sản phẩm làm được gì
* Đầu ra: tài liệu thiết, mô tả về cách sản phẩm làm được điều này
* Mục đích: Chuyển tài liệu phân tích từ dạng đặc tả nghiệp vụ hệ thống sang dạng có thể cài đặt và kiểm thử được
* Trong quy trình thiết kế, các tạo phẩm này được lặp đi lặp lại và tăng dần cho đến khi chúng ở định dạng mà các lập trình viên có thể sử dụng
* Gồm 3 hoạt động chính
* Thiết kế kiến trúc
* Quá trình phân tách modul được phát triển. Nếu sử dụng mô hình hướng đối tượng thì bước này đã được thực hiện ở quy trình phân tích rồi ( do một lớp là một modul mà bước đầu tiên ở quy trình phân tích là xác định các lớp)
* Thiết kế chi tiết – thiết kế modul: Các thuật toán cụ thể được chọn và cấu trúc dữ liệu được chọn
* Thử nghiệm thiết kế: được thực hiện đồng thời trong suốt quá trình thiết kế
* Mục tiêu: để xác minh rằng các thông số kỹ thuật đã được tích hợp chính xác và đầy đủ vào thiết kế cũng như để đảm bảo tính đúng đắn của bản thân thiết kế
* Điều quan trọng là bất kỳ lỗi nào trong thiết kế đều được phát hiện trước khi bắt đầu viết mã -> giảm chi phí
* Việc kiểm tra bao gồm tất cả các loại giao dịch có thể nên được lên lịch.
* Các kỹ thuật thiết kế: kỹ thuật hướng hoạt động, kỹ thuật hướng dữ liệu và kỹ thuật hướng đối tượng.
* Các bước thực hiện của kỹ thuật hướng đối tượng: 2 bước chính là hoàn thiện sơ đồ lớp và thực hiện chi tiết hoạt động.
* Hoàn thiện sơ đồ lớp có được trong pha phân tích -> sơ đồ lớp chi tiết
* Định nghĩa kiểu thuộc tính cho lớp: tên thuộc tính -> xác định kiểu dữ liệu
* Định nghĩa khuôn mẫu các phương thức cho lớp: Có 3 nguyên tắc
* Nguyên tắc 1: Che giấu thông tin:
* Nguyên tắc 2: Nếu có nhiều đối tượng gọi đến một hành động của một đối tượng khác thì phương thức để thực hiện hành động nên gán cho lớp của một đối tượng
* Nguyên tắc 3: thiết kế hướng đến trách nhiệm.
* Thiết kế chi tiết hoạt động của các lớp, các phương thức của nó
* Một sản phẩm phần mềm lớn sẽ dễ phát triển hơn nếu nó có thể được phân tách thành các gói tương đối độc lập. Phân tách một sản phẩm phần mềm thành các gói là một ví dụ về chia để trị. Có 2 lý do:
* Việc triển khai một số hệ thống con nhỏ sẽ dễ dàng hơn so với một hệ thống lớn. Nghĩa là, chia nhỏ một sản phẩm phần mềm thành các hệ thống con là một ví dụ khác về chia để trị
* Nếu các hệ thống con được triển khai thực sự tương đối độc lập, thì chúng có thể được triển khai bởi các nhóm lập trình làm việc song song. Điều này dẫn đến toàn bộ sản phẩm phần mềm được chuyển giao sớm hơn.

**2: Các kỹ thuật áp dụng**

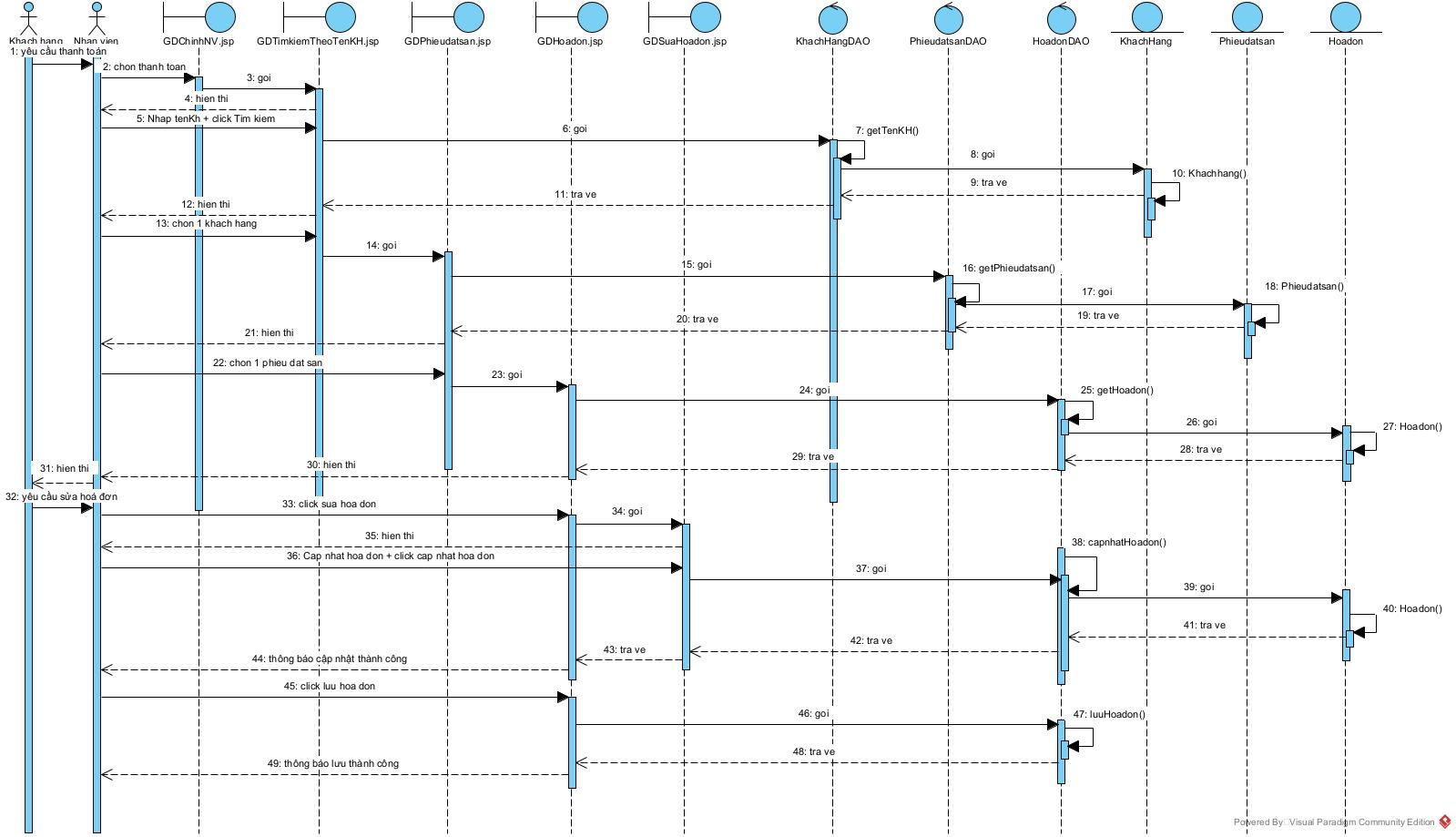
* Kỹ thuật hướng hành động:2 kỹ thuật
* Phân tích luồng dữ liệu (DFA)
* Một kỹ thuật thiết kế cổ điển để đạt được các modul có độ gắn kết cao.
* Được sử dụng kết hợp với hầu hết các kỹ thuật phân tích.
* Đầu vào là một sơ đồ luồng dữ liệu
* Phân tích giao dịch: Giao dịch là một hoạt động từ quan điểm của người dùng sản phẩm, chẳng hạn như “xử lý yêu cầu” hoặc “in danh sách các đơn đặt hàng hôm nay”
* Thiết kế kiến ​​trúc theo hai thành phần:phân tích và điều phối
* Đối với mỗi tập hợp các hoạt động liên quan: Thiết kế một mô-đun cơ bản và khởi tạo nó nhiều lần nếu cần.
* Kỹ thuật hướng dữ liệu
* Nguyên tắc cơ bản đằng sau: là thiết kế sản phẩm theo cấu trúc của dữ liệu mà nó hoạt động. Đầu tiên cấu trúc của dữ liệu được xác định -> mỗi thủ tục được đưa ra cấu trúc giống như dữ liệu mà nó hoạt động.
* Không được phổ biến
* Kỹ thuật hướng đối tượng
* Thể hiện của các lớp và các lớp con được trích xuất trong quá trình phân tích hướng đối tượng.

**3: Các kết quả (artifact) cần đạt**

* Xây dựng một quy trình thiết kế: ở đây chọn thiết kế hướng đối tượng
* Biểu đồ lớp chi tiết
* Biểu đồ tương tác
* Biểu đồ tuần tự

**4: Áp dụng cho case study Thanh toán:**

* Biểu đồ tuần tự:



**PHA THỰC THI**

**1: Các công việc cần thực hiện trong luồng công việc**

* Thực hiện quy trình thực hiện.
* Lựa chọn ngôn ngữ lập trình
* Nguyên tắc kỹ thuật phần mềm có thể dẫn đến code chất lượng tốt hơn.
* Thực hành lập trình tốt: Sử dụng tên biến có tính nhất quán và có ý nghĩa, chú thích code, sử dụng tham số, bố cục code tăng khả năng đọc
* Chuẩn hoá code:
* Tái sử dụng code:
* Tích hợp: tích hợp từ trên xuống, tích hợp từ dưới lên, tích hợp sandwich, tích hợp các sản phẩm hướng đối tượng.
* Nguyên tắc phân chia và code modul
* Một method(modul) chỉ tối đá 30 -50 câu lệnh
* Nếu nhiều hơn thì nên tách thành một modul con
* Thực hiện kiểm thử đơn vị hộp đen, hộp kính(nêu ở phần kỹ thuật)
* Cleanroom: là một cách tiếp cận khác đối với phát triển phần mềm
* Một số vấn đề có khả năng cao xảy ra khi kiểm thử các đối tượng: lớp, đối tượng, chạy từng trường hợp thử nghiệm với các đối tượng, các đối tượng điểm hình, các phương thức kế thừa,…
* Khía cạnh khác của kiểm thử đơn vị: phân tích chi phí – lợi nhuận, phân tích rủi ro, các kỹ thuật thống kê
* Thực hiện kiểm thử tích hợp: là việc kiểm thử khi có một modun mã mới được thêm vào nhóm các modul đã được kiểm thử
* Tiến hành kiểm thử sản phẩm :
* Cho phần mềm COTS: alpha, beta
* Đối với phần mềm tuỳ chỉnh: đảm bảo phần mềm chuyển qua kiểm thử chấp nhận
* Kiểm thử chấp nhận: khách hàng xác định liệu phần mềm thoả mãm những đặc tả của nó. 4 thảnh phần chính: tính chính xác, tính mạnh mẽ, hiệu năng và tài liệu.

**2:Các kỹ thuật được áp dụng**

* Các kỹ thuât tính hợp:
* Top – down: Tích hợp từ trên xuống
* Ưu điểm: hỗ trợ cách ly lỗi, giúp phát hiện sớm các lỗi thiết kế lớn
* Nhược điểm: các modul có khả năng tái sử dụng không được kiểm thử một cách thích đáng, các modul thấp hơn(mức hành động) không được kiểm thử thường xuyên
* Bottom – up: Tích hợp từ dưới lên
* Ưu điểm: các modul hoạt động được kiểm thủ kỹ lưỡng, các code có khả năng tái sử dụng có tính đầy đủ(giải quyết được yếu điểm của Top -down)
* Nhược điểm: thiết kế chính sẽ bị phát hiện muộn. các thành phần logic được tạo ra cuối cùng, do đó nếu có nếu có lỗi thiết kế lớn, sẽ được xử lý ở cuối => tốn kém chi phí và thời gian
* Sandwich :các modul logic được tích hợp từ trên xuống, các modul hoạt động tích hợp từ dưới lên, tích hợp theo cả 2 chiều trên xuống và dưới lên: có được điểm mạnh của cả 2 kỹ thuật trên
* Kỹ thuật kiểm thử hộp đen:
* Kiểm thử tương đương và phân tích các giá trị biên
* Kiểm thử chức năng
* Kỹ thuật kiểm thử hộp kính:
* Phủ dòng lệnh
* Phủ nhánh
* Phủ đường dẫn
* Chuỗi mã tuyến tính
* Phủ đường dẫn sử dụng tất cả định nghĩa

**3: Các kết quả (artifact) cần đạt**

* Lựa chọn được ngôn ngữ lập trình
* Nắm được các chuẩn lập trình cơ bản
* Các kỹ thuật tích hợp: trên xuống, dưới lên ,sandwich
* Nắm đươc kỹ thuật kiểm thử hộp đen, hộp trắng
* Kiểm thử tích hợp
* Kiểm thử sản phẩm
* Kiểm thử chấp nhận