**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**------------- o0o -------------**



**BÁO CÁO**

**PHƯƠNG PHÁP MÔ HÌNH HÓA**

***ĐỀ TÀI***

**MÔ HÌNH HÓA HỆ THỐNG SỬ DỤNG UML**

**MÔ HÌNH HÓA USE CASE**

**Giảng viên hướng dẫn:**

**PGS. TS. Vũ Thanh Nguyên**

**Lớp: SE101.J11.PMCL**

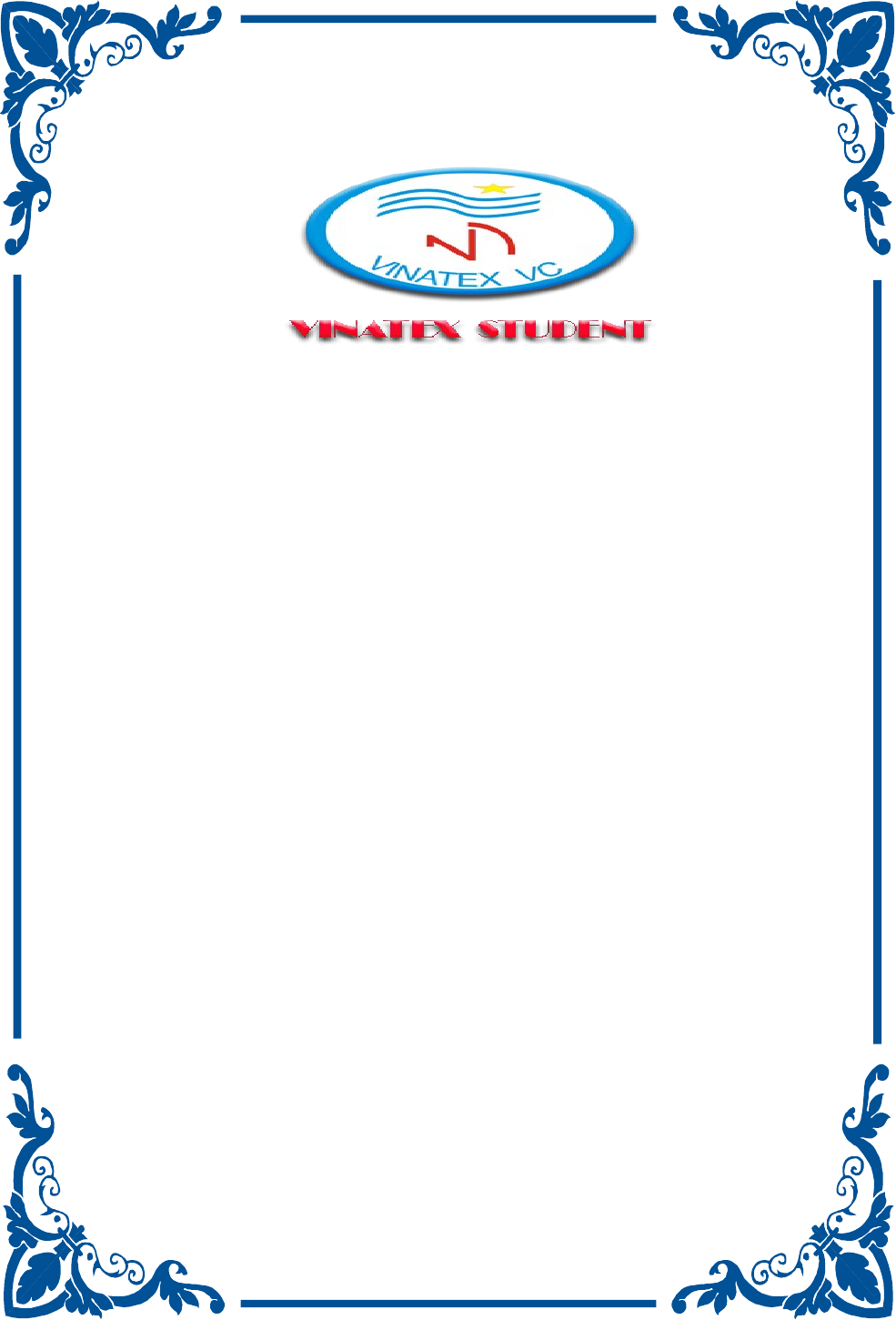
**Nhóm sinh viên thực hiện:**

**Nguyễn Huỳnh Sáng – 16521022**

**Phan Hoàng Nhân – 16520873**

**Hồng Gia Lập – 16520654**

*TP.HCM, 25/12/2018*



**NHẬN XÉT CỦA GIẢNG VIÊN**

**LỜI CẢM ƠN**

Lời đầu tiên, cả nhóm xin gửi lời cảm ơn đến thầy PGS.TS. Vũ Thanh Nguyên. Thầy đã nhiệt tình giảng dạy trên lớp, hỗ trợ những thông tin cần thiết và giải đáp những thắc mắc cho nhóm và các bạn trong suốt quá trình thực hiện đề tài. Thông qua những bài giảng trên lớp về môn học của Thầy, các thành viên trong nhóm đã tiếp thu được nhiều kiến thức hữu ích liên quan như các khái niệm và nguyên lý về mô hình hóa và việc sử dụng các mô hình trong các biểu diễn bao gồm biểu diễn dữ liệu, biểu diễn thông tin, biểu diễn tri thức, biểu diễn vấn đề và lời giải, và biểu diễn các hệ thống, các phương pháp hệ thống hóa để phát triển phần mềm thông qua phương pháp mô hình hóa hệ thống. Qua đó, giúp chúng em có thể hiểu và nắm bắt được quy trình để phát triển một hệ thống hay chương trình phần mềm hoàn chỉnh.

Đề tài “Quản Lý Nhà Sách” hoàn thành là kết quả của quá trình nghiên cứu nghiêm túc của các thành viên của cả nhóm trong quá trình học tập và tiếp thu kiến thức dưới sự hướng dẫn tận tình của giảng viên hướng dẫn và các tài liệu được giảng viên hướng dẫn chia sẻ qua các kênh giao tiếp trong suốt thời gian giảng dạy.

Đồng thời nhóm cũng muốn cảm ơn các anh chị khóa trên, đặc biệt là anh chị trong khoa đã chia sẽ kinh nghiệm quý báu về môn học cũng như những kiến thức liên quan. Cũng xin cảm ơn bạn bè đã tạo điều kiện thuận lợi, mọi người đã đưa ra nhận xét và góp ý chân thành, vô cùng quý giá. Những người đã động viên, hỗ trợ nhóm hoàn thành đề tài.

*Tp. Hồ Chí Minh, tháng 12 năm 2018*

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc533522888)

[THÔNG TIN NHÓM 2](#_Toc533522889)

[PHẦN 1: GIỚI THIỆU VỀ PHƯƠNG PHÁP HƯỚNG ĐỐI TƯỢNG 3](#_Toc533522890)

[CHƯƠNG 1: TỔNG QUAN VỀ PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG 3](#_Toc533522891)

[I. Lịch sử 3](#_Toc533522892)

[II. Khái niệm, định nghĩa 3](#_Toc533522893)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HƯỚNG ĐỐI TƯỢNG 4](#_Toc533522894)

[I. Phân tích thiết kế hướng đối tượng (OOAD) 4](#_Toc533522895)

[II. Phân tích hướng đối tượng 9](#_Toc533522896)

[III. Nguyên tắc thiết kế hướng đối tượng 10](#_Toc533522897)

[CHƯƠNG 3: QUY TRÌNH PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG 11](#_Toc533522898)

[I. Xác định yêu cầu 11](#_Toc533522899)

[II. Phân tích hệ thống 13](#_Toc533522900)

[III. Thiết kế hệ thống 14](#_Toc533522901)

[IV. Lập trình, mã hóa chương trình 15](#_Toc533522902)

[V. Kiểm định hệ thống 15](#_Toc533522903)

[VI. Vận hành và bảo trì hệ thống 16](#_Toc533522904)

[PHẦN 2: CÁC BƯỚC XÂY DỰNG MÔ HÌNH HÓA USE CASE 16](#_Toc533522905)

[CHƯƠNG 1: XÁC ĐỊNH CÁC TÁC NHÂN HỆ THỐNG: 16](#_Toc533522906)

[CHƯƠNG 2: XÁC ĐỊNH USE CASE VÀ CÁC MỐI QUAN HỆ: 17](#_Toc533522907)

[PHẦN 3: MÔ HÌNH HÓA PHẦN MỀM “QUẢN LÝ NHÀ SÁCH” 25](#_Toc533522908)

[CHƯƠNG 1: MÔ TẢ NGHIỆP VỤ 25](#_Toc533522909)

[I. Danh sách các yêu cầu 25](#_Toc533522910)

[II. Danh sách các biểu mẫu và quy định 25](#_Toc533522911)

[III. Bảng trách nhiệm yêu cầu nghiệp vụ 28](#_Toc533522912)

[CHƯƠNG 2: KIẾN TRÚC HỆ THỐNG 30](#_Toc533522913)

[I. Kiến trúc hệ thống 30](#_Toc533522914)

[II. Sơ đồ Use Case tổng quát 33](#_Toc533522915)

[CHƯƠNG 3: XÂY DỰNG BIỂU ĐỒ UML 35](#_Toc533522916)

[I. Đăng nhập 35](#_Toc533522917)

[II. Đổi mật khẩu 37](#_Toc533522918)

[III. Đăng xuất 41](#_Toc533522919)

[iv. Sửa thông tin sách 43](#_Toc533522920)

[v. Sửa thông tin khách hàng 46](#_Toc533522921)

[vi. Tra cứu Sách 49](#_Toc533522922)

[vii. Tra cứu Khách hàng 51](#_Toc533522923)

[viii. Thêm Sách 53](#_Toc533522924)

[ix. Thêm Khách hàng 55](#_Toc533522925)

[x. Lập Hóa Đơn 57](#_Toc533522926)

[xi. Lập phiếu nhập sách 60](#_Toc533522927)

[xii. Lập phiếu thu tiền 63](#_Toc533522928)

[xiii. Báo cáo tồn 65](#_Toc533522929)

[xiv. Báo cáo nợ công 67](#_Toc533522930)

[xv. Thay đổi quy định 69](#_Toc533522931)

[xvi. Thêm Người dùng 71](#_Toc533522932)

[CHƯƠNG 3: THIẾT KẾ DỮ LIỆU 73](#_Toc533522933)

[I. Sơ đồ 73](#_Toc533522934)

[II. Diễn giải 74](#_Toc533522935)

[CHƯƠNG 4: THIẾT KẾ GIAO DIỆN 79](#_Toc533522936)

[I. Màng hình chính 79](#_Toc533522937)

[II. Màng hình “Lập phiếu nhập sách” 80](#_Toc533522938)

[III. Màng hình “Lập hóa đơn bán sách” 83](#_Toc533522939)

[IV. Màng hình “tra cứu” 87](#_Toc533522940)

[V. Màng hình “Lập phiếu thu tiền” 90](#_Toc533522941)

[VI. Màng hình “lập báo cáo tồn” 94](#_Toc533522942)

[VII. Màng hình “Báo cáo công nợ” 96](#_Toc533522943)

[VIII. Màng hình “Thay đổi quy định” 99](#_Toc533522944)

[IX. Màng hình đăng nhập 101](#_Toc533522945)

[CHƯƠNG 5: TỔNG KẾT 103](#_Toc533522946)

[CHƯƠNG 6: TÀI LIỆU THAM KHẢO 105](#_Toc533522947)

# **LỜI MỞ ĐẦU**

Đứng trước cuộc Cách mạng Công nghiệp lần thứ IV, nhu cầu ứng dụng các sản phẩm thành tựu của công nghệ, kĩ thuật vào các mặt của đời sống, công việc ngày càng được chú trọng. Việc giải các bài toán, các vấn đề phát sinh ra trong quá trình lao động, quản lí – từ đơn giản đến phức tạp, luôn là mối quan tâm hàng đầu của cá nhân, tập thể.

Sự ra đời của các sản phẩm phần mềm được sáng tạo, sản xuất ra nhằm giải quyết các vấn đề của con người theo một cách hiệu quả, tích cực và tiết kiệm tiềm năng, thời gian và chi phí của người sử dụng. Một trong những vấn đề nan giải chính là các vấn đề, bài toán về quản lí. và lưu trữ thông tin, dữ liệu – cụ thể hơn như quản lí các dữ liệu của một nhà sách.

Phần mềm Quản lí Nhà sách ra đời giúp nhân viên của nhà sách thực hiện tốt hơn công việc của mình, các thao tác lưu trữ, tìm kiểm, xử lí và báo cáo thông tin đều được đơn giản hóa và tiện dụng cho người thủ thử quản lí. Tất cả đều được phần mềm hóa, tin học hóa – mang lại cách giải quyết các nhu cầu của người dùng nhanh chóng và chính xác nhất.

Với sự hướng dẫn tận tình của Thầy, nhóm chúng em đã hoàn thành đồ án cũng như báo cáo này. Tuy đã cố gắng hết sức tìm hiểu, phân tích thiết kế và cài đặt hệ thống nhưng sự hạn chế về kinh nghiệm cũng như trình độ hiểu biết nên chắc chắn rằng sẽ không tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía thầy nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện các đề tài khác trong tương lai.

# **THÔNG TIN NHÓM**

1. **– Thông tin nhóm**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Email | Vai trò |
| 16521022 | Nguyễn Huỳnh Sáng | 16521022@gm.uit.edu.vn | Trưởng nhóm |
| 16520873 | Phan Hoàng Nhân | 16520873@gm.uit.edu.vn | Thành viên |
| 16520654 | Hồng Gia Lập | 16520654@gm.uit.edu.vn | Thành viên |

1. **– Phương thức làm việc**

**Quy trình**

* Thống nhất đề tài đồ án
* Phân công nhiệm vụ cho từng thành viên
* Tìm kiếm tài liệu, lên kế hoạch cho các mốc thời gian cho đồ án
* Tiến hành các buổi họp nhóm, trao đổi thông tin, quy trình, bàn bạc kế hoạch thực hiện, phân công nhiệm vụ cho mỗi thành viên và thống nhất thời gian dealine
* Hoàn thành đồ án, cho ra sản phẩm hoàn chỉnh
* Đánh giá và thử nghiệm sản phẩm, hoàn thành báo cáo đồ án

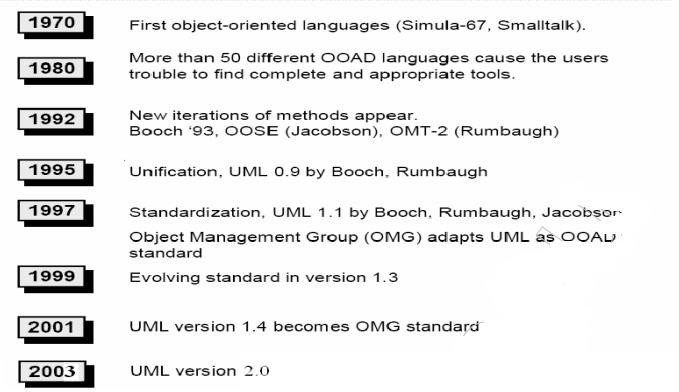
**Công cụ**

* Visual Studio
* SQL Server
* Github
* Microsoft Visio

# **PHẦN 1: GIỚI THIỆU VỀ PHƯƠNG PHÁP HƯỚNG ĐỐI TƯỢNG**

## CHƯƠNG 1: TỔNG QUAN VỀ PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

### Lịch sử



### Khái niệm, định nghĩa

Lập trình hướng đối tượng (Object-oriented programing, viết tắt là OOP) là một mẫu hình lập trình dựa trên khái niệm “Công nghệ đối tượng” mà trong đó, các đối tượng chứa đựng các thuộc tính, và phương thức. OOP giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, có nhiều ý kiến cho rằng OOP dễ tiếp thu và thực hiện hơn cho những người mới học về lập trình hơn là những phương pháp khác.

Một cách giản lược, đây là khái niệm và sự nỗ lực nhằm giảm nhẹ các thao tác viết mã cho người lập trình, cho phép họ tạo ra các ứng dụng và các yếu tố bên ngoài có thể tương tác với các chương trình đó giống như là tương tác vật lý.

Những đối tượng trong một ngôn ngữ OOP là các kết hợp giữa mã và dữ liệu mà chúng được nhìn nhận như là một đơn vị duy nhất. Mỗi đối tượng có một tên riêng biệt và tất cả các tham chiếu đến đối tượng đó được tiến hành thông qua tên của nó. Như vậy, mỗi đối tượng có khả năng nhận vào các thông báo, xử lý dữ liệu (bên trong của nó), và gửi ra hay trả lời đến các đối tượng khác hay đến môi trường.

## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HƯỚNG ĐỐI TƯỢNG

### Phân tích thiết kế hướng đối tượng (OOAD)

* 1. **Khái niệm**

Trong những năm gần đây, phương thức lập trình hướng đối tượng đã trở nên rất phổ biến trên thị trường lập trình phần mềm và UML cũng đã trở thành ngôn ngữ mô hình hóa phổ biến trong sản xuất phần mềm.

Đối với ngành kỹ thuật phần mềm, để sản xuất được một phần mềm đạt yêu cầu thì người ta chia quá trình phát triển nó ra nhiều giai đoạn:



Thu thập,

phân tích

yêu cầu

Triển khai

và Bảo trì

Quá trình

phát triển

Phân tích,

thiết kế hệ thống

Kiểm thử

Coding

Trong đó, giai đoạn phân tích thiết kể bao giờ cũng là giai đoạn khó khăn và phức tạp nhất. Giai đoạn này giúp chúng ta hiểu rõ yêu cầu đặt ra, xác định giải pháp, mô tả chi tiết giải pháp. Để phân tích và thiết kế một phần mềm thì có rất nhiều phương pháp để thực hiện, một trong những cách đó là xem hệ thống gồm những đối tượng sống trong đó và tương tác với nhau. Việc mô tả được tất cả các đối tượng và sự tương tác của chúng sẽ giúp chúng ta hiểu rõ hơn về hệ thống để có thể dễ dàng cài đặt. Phương thức này gọi là **Phân tích thiết kế hướng đối tượng.**

* 1. **Khái niệm về UML (Unified Modeling Language)**

UML là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau, đồng thời dùng để thi công hệ thống, thuyết phục khách hàng và các nhà đầu tư.

* 1. **Tại sao lại là OOAD và UML?**

OOAD cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung dung thể hiện. Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó, vì vậy chúng thường đi đôi với nhau.

* 1. **OOAD sử dụng UML**

UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng… OOAD sử dụng UML bao gồm các thành phần sau:

* View
* Diagram
* Notations
* Mechanisms

#### View

Mỗi view được ví như “Thầy bói xem voi”, thông qua đó chúng ta có thể thấy được sự thể hiện rõ rệt của hệ thống ở một khía cạnh mà không thể nhìn thấy toàn bộ hệ thống. Chính vì thế trong xây dựng có bản vẽ kiến, bản vẽ kết cấu thì trong phần mềm, OOAD sử dụng AML có các View sau:

Logical View

Component View

*Funtionality*

*Software Management, Reuse,*

*Portability*

**Use Case View**

*Understandability, Usability*

Process View

Deployment View

Performance, Availability, Fault Tolerance

Performance, Availability, Fault Tolerance,

Scalability, Delivery and Installation

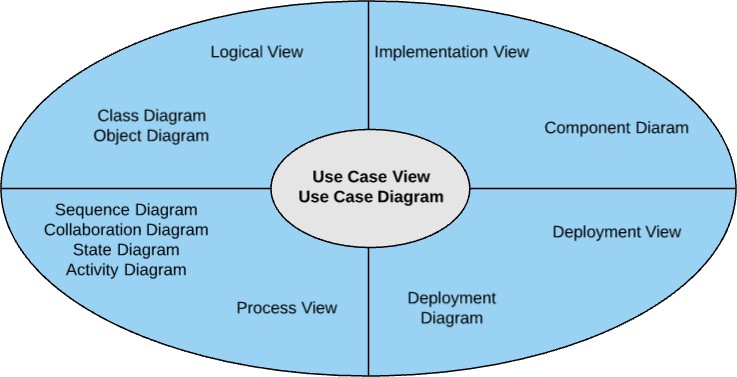
**Hình 2. Các View trong OOAD sử dụng UML**

Trong đó:

* **Use Case View:** Cung cấp góc nhìn về các ca sử dụng giúp chúng ta hiểu hệ thống có gì? Ai có thể sử dụng và sử dụng nó như thế nào.
* **Logical View:** Được sử dụng để xem xét các phần tử bên trong hệ thống và mối quan hệ, sự tương tác giữa chúng để thực hiện các chức năng mong đợi của hệ thống.
* **Process View:** Chia hệ thống thành các tiến trình và luồng, mô tả việc đồng bộ hóa và các xử lý đồng thời. Từ đó người dùng có thể phát triển và tích hợp hệ thống.
* **Component View:** Cũng là một góc nhìn về cấu trúc giúp chúng ta có thể hiểu được cách phân bổ và sử dụng các thành phần trong hệ thống ra sao.
* **Deployment View:** Chỉ ra cấu hình phần cứng mà hệ thống sẽ chạy trên đó. Nó thể hiện sự phân tác, cài đặt các phần mà tạo nên kiến trúc vật lý của hệ thống.

#### Diagram

Diagram là một biểu đồ mà trong đó bao gồm các ký hiệu phần tử mô hinh hóa được sắp xếp để minh họa mộ thành phần hoặc một khía cạnh cụ thể của hệ thống.



**Hình 2. Các bản vẽ trong OOAD sử dụng UML**

Trong đó,

* Use Case Diagram: Sơ đồ mô tả về cách sử dụng của hệ thống. Bản vẽ này sẽ giúp chúng ta biết được ai sử dụng hệ thống, hệ thống bao gồm những chức năng gì? Nhờ đó ta sẽ hểu được yêu cầu của hệ thống cần được xây dựng
* Class Diagram: sơ đồ này mô tả cấu trúc của hệ thống, tức là hệ thống được cấu tạo từ những thành phần nào. Đồng thời nó cũng mô tả khía cạnh tĩnh của hệ thống.
* Object Diagram: Tương tự như Class Diagram nhưng nó mô tả đến đối tượng thay vì lớp.
* Sequence Diagram: Mô tả sự tương tác của các đối tượng trong hệ thống với nhau và được mô tả theo một cách tuần tự từng bước tương tác theo thời gian.
* Collaboration Diagram: Tương tự như Sequence Diagram nhưng ở đây, nó nhấn mạnh về sự tương tác thay vì tuần tự theo thời gian.
* State Diagram: Sơ đồ mô tả sự thay đổi trạng thái của một đối tượng, nó được dùng để theo dõi các đối tượng có trạng thái được thay đổi nhiều trong hệ thống.
* Activity Diagram: Sơ đồ mô tả các hoạt động của một đối tượng, thường được sử dụng để hiểu về nghiệp vụ của hệ thống.
* Component Diagram: Sơ đồ mô tả về việc bố trí các thành phần của hệ thống cũng như cách sử dụng các thành phần đó.
* Deployment Diagram: Sơ đồ mô tả việc triển khai hệ thống như kết nối, cài đặt, hiệu năng của hệ thống.

#### Notations

Notation là các ký hiệu dùng để vẽ, được hiểu như là từ vựng trong ngôn ngữ tự nhiên.



**Hình 3. Ký hiệu về Use Case**

**Hình 4. Ký hiệu về Class**



**Hình 5. Ký hiệu về Actor**

* + 1. Mechanisms (Rules)

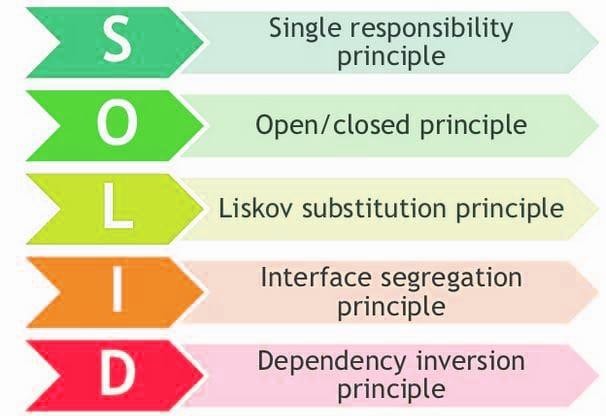
Mechanisms là các qui tắc để lập nên bản vẽ, mỗi bản vẽ có qui tắc riêng và chúng ta phải nắm được cụ thể để có thể tạo nên các các bản vẽ thiết kế đúng đắn nhất.

### Phân tích hướng đối tượng

Áp dụng kỹ thuật mô hình đối tượng để phân tích các yêu cầu, chức năng cho hệ thống.

* 1. **Nhiệm vụ:**
* Thiết lập cách nhìn tổng quan về hệ thống.
* Liệt kê các nhiệm vụ mà hệ thống cần thực hiện.
* Đưa ra hướng giải quyết bài toán.
* Phát triển những vấn đề liên quan trong miền quan tâm của bài toán.
  1. **Các bước thực hiện**
* Bước 1: Mô hình Use Case (Xây dựng mô hình chức năng của sản phầm phần mềm nhìn từ quan điểm của người sử dụng)
* Bước 2: Mô hình lớp (Biểu diễn các lớp, các thuộc tính và mối quan hệ giữa các lớp.
* Bước 3: Mô hình động (Biểu diễn các hoạt động liên quan đến 1 lớp hay lớp con)

### Nguyên tắc thiết kế hướng đối tượng



* 1. **Open/ closed priciple (Nguyên tắc đóng/ mở)**

Các thực thể phần mềm nên xây dựng theo hướng mở cho việc mở rộng và đóng gói đối với việc sửa đổi.

Nguyên lý Mở - Đóng giúp cho việc thiết kế và xây dựng chương trình ổn định và có thể tái sử dụng.

* 1. **Single responsbility principle (Nguyên tắc đơn nhiệm)**

Nguyên tắc đơn nhiệm phát biểu rằng: Mỗi class hay method chỉ nên có một và chỉ một công năng. Những nguyên tắc này thường rất dễ vi phạm, ví dụ như một class User chịu trách nhiệm trong quá nhiều công việc trong ứng dụng. Những class như thế được gọi là “Gob object”.

* 1. **Liskov substitution principle**

Nếu một class có sử dụng một implementation của một interface thì nó phỉ được thay thế dễ dàng bởi các implementation của interface đó mà không cần sửa gì thêm.

* 1. **Interface segregation principle (Nguyên tắc tách rời giao diện)**

Implementation của một interface không nên bị phụ thuộc vao những methods mà nó không sử dụng nhiều. Điều này có nghĩa là các interface phải được sắp xếp và phân chia hợp lý. Thay vì có một FAT interface chứa tất cả các methods cần được thi công thì chúng ta nên chia nhỏ ra.

* 1. **Dependenct inversion principle (Nguyên tắc ngịch đảo)**

Hight – level code không nên phụ thuộc vào low – level code. Thay vào đó hight – level nên phụ thuộc vào một abstraction mà hỗ trợ tương tác với low – level code đó nhưng không được phụ thuộc chi tiết của low – level code đó.

## CHƯƠNG 3: QUY TRÌNH PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

Quá trình phát triển phần mềm hướng đối tượngn được xác định thông qua tập hoạt động cần thực hiện để chuyển đổi yêu cầu của khách hàng thành hệ thống phần mềm.

Có 6 bước cần thực thiện trong quá trình phát triển phần mềm:

* Bước 1: Xác định và đặc tả các yêu cầu
* Bước 2: Phân tích hệ thống
* Bước 3: Thiết kế hệ thống
* Bước 4: Lập trình, mã hóa chương trình
* Bước 5: Kiểm định hệ thống
* Bước 6: Vận hành và bảo trì hệ thống

1. **Xác định yêu cầu**

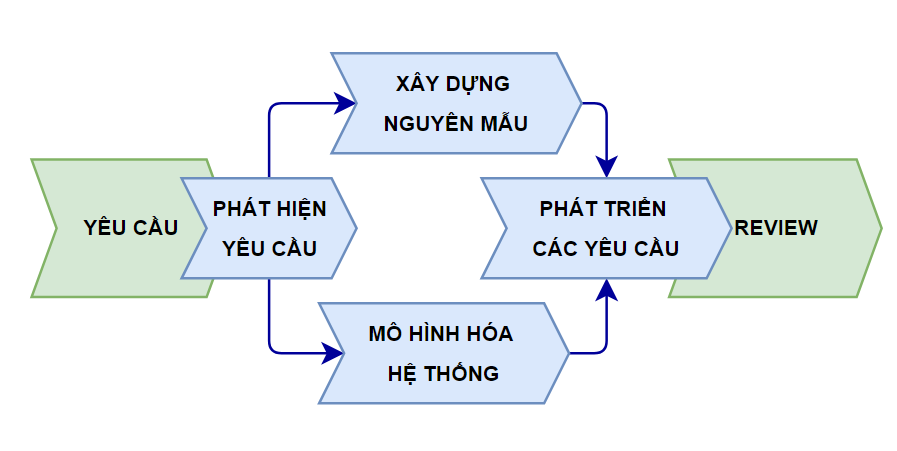
Yêu cầu phần mềm là tất cả các yêu cầu về phần mềm cho khách hàng nêu ra, bao gồm: các chức năng của phần mềm, hiệu năng và các yêu cầu thử đánh giá.

Yêu cầu phần mềm được chia làm bốn loại theo thành phần của phần mềm:

* Yêu cầu về phần mềm
* Yêu cầu về phần cứng
* Yêu cầu về dữ liệu
* Yêu cầu về con người

Mục đích cơ bản của nắm bắt yêu cầu là hướng sự phát triển đến một hệ thống đúng đắn. Kết quả của nắm bắt yêu cầu đồng thời cũng giúp người quản lý dự án lập ké hoạch các bước lặp và các xuất phẩm cho khách hàng.

Quy trình xác định yêu cầu phần mềm bao gồm:



#### Phát hiện các yêu cầu phần mềm

* + Xác định các phương pháp sử dụng để phát hiện các yêu cầu phần mềm như phỏng vấn, làm việc nhóm.
  + Tìm kiếm nhân sự có hiểu biết sâu sắc và chi tiết về hệ thống giúp chúng ta xác định yêu cầu phần mềm
  + Xác định môi trường kĩ thuật, các ràng buộc lĩnh vực.
  + Thu hút sự tham gia của nhiều chuyên gia và khách hàng để có các quan điểm xem xét phần mềm từ nhiều khía cạnh khác nhau

#### Phân tích các yêu cầu phần mềm và thương lượng với khách hàng

* + Phân loại các yêu cầu và sắp xếp chúng theo các nhóm liên quan.
  + Khảo sát, thẩm định từng yêu cầu theo các tính chất phù hợp đầy đủ, rõ ràng và không trùng lặp
  + Cần thiết phải thẩm định từng yêu cầu, đánh giá rủi có thể xảy ra với nó.
  + Giải quyết bất đồng với yêu cầu phần mềm với khách hàng trên cơ sở thảo luận và thương lượng với yêu cầu đặt ra.

#### Đặc tả yêu cầu phần mềm

* + Việc đặc tả các yêu cầu phần mềm sử dụng các công cụ như: mô hình hóa, mô hình toán học để làm một hình thức, tập hợp các kịch bản sử dụng các nguyên mẫu.
  + Yêu cầu của đặc tả bao gồm: Dễ hiểu với ngươi dùng, có quy ước khi mô tả, theo quy cách từ trên xuống và cuối cùng là dễ triển khai.

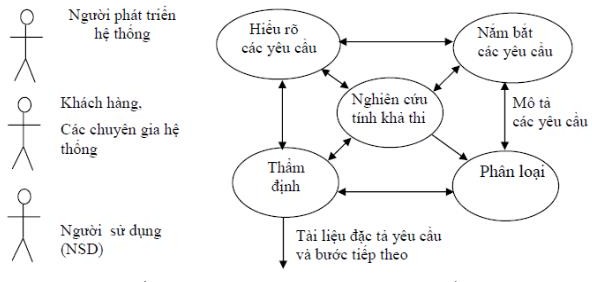
1. **Phân tích hệ thống**

Từ yêu cầu của khách hàng, chúng ta xác định được các mục tiểu của phần mềm cần phát triển. Những yêu cầu chức năng đó là những gì hệ thống phải thực hiện, việc xác định đúng và đầy đủ các yêu cầu là nhiệm vụ rất quan trọng. Sự cần thiết phải đặc tả chi tiết các yêu cầu là nhiệm vụ quan trọng, sử dụng các UML để đặc tả các yêu cầu hệ thống.

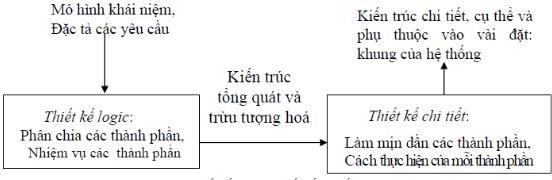
Thông qua đặc tả chi tiết của từng yêu cầu, chúng ta lấy đó làm cơ sở để trao đổi với người sử dụng, để thảo luận giữa các nhóm thành viên trong dự án phát triển phần mềm về những gì mà hệ thống sẽ phải thực hiện. Đồng thời làm căn cứ cơ bản để kiểm tra, thử nghiệm trong các bước tiếp theo của quá trình phát triển phần mềm.

Các mục tiêu trong phân tích hệ thống:

* Xác định và hiểu rõ miền, phạm vi của bài toán.
* Nắm bắt các yêu cầu.
* Phân loại
* Thẩm định



1. **Thiết kế hệ thống**



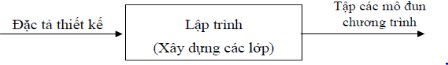
Đây là một trong các giai đoạn trong quá trình phát triển phần mềm, trong đó hệ thống được tổ chức thành lập các đối tượng tương tác với nhau và mô tả được cách hệ thống thực thi nhiệm vụ của bài toán ứng dụng.

Trong giai đoạn này, việc xây dựng các thiết kế chi tiết mô tả các thành phần của hệ thống ở mức cao hơn khâu phân tích, thiết kế giao diện để phục vụ cho việc cài đặt. Nghĩa là các lớp đối tượng được định nghĩa chi tiết gồm đầy đủ các thuộc tính, các thao tác phục vụ cho việc cài đặt bằng ngôn ngữ lập trình hướng đối tượng ở bước kế tiếp. Đồng thời, chúng ta phải đưa ra được trọng tâm của hệ thống để đảm bảo cho hệ thống có thể thay đổi, có tính mở, dễ bảo mật, thân thiện với người sử dụng,...

Những kết quả trên được thể hiện trong các biểu đồ: Biểu đồ lớp (chi tiết), biểu đồ hành động, biểu đồ thành phần và biểu đồ triển khai. Tất cả các kết quả thiết kế phải được ghi lại thành các hồ sơ, tài liệu cho hệ thống. Trong các tài liệu thiết kế, chúng ta cần phải mô tả được các thành phần và nhiệm vụ của nó trong hệ thống.

1. **Lập trình, mã hóa chương trình**

Trong giai đoạn này, mỗi thành phần đã được thiết kế sẽ được lập trình bằng ngôn ngữ lập trình hướng đối tượng được lựa chọn thành những mô đun chương trình (chương trình con). Mỗi mô đun này sẽ được kiểm định hoặc thử nghiệm theo các tài liệu đặc tả của giai đoạn thiết kế.



1. **Kiểm định hệ thống**

Kiểm thử là một quá trình liên tục, xuyên suốt mọi giai đoạn trong quá trình phát triển nhằm đảm bảo rằng phần mềm thỏa mãn các yêu cầu thiết kế, thẩm định xem những yêu cầu đó có đáp ứng các nhu cầu của người dùng hay không.

Có hai loại kỹ thuật kiểm định phần mềm chính là kỹ thuật “Hộp đep” và kỹ thuật “Hộp trắng”. Trong đó, kỹ thuật kiểm thử “Hộp đen” còn được gọi là kỹ thuật hướng dữ liệu vào/ra. Nó được sử dụng để kiểm tra các đặc tả chức năng của thiết kế và không quan tâm đến cấu trúc bên trong của hệ thống. Nếu các đầu ra không đúng như mong muốn thì kiểm thử thành công trong việc phát hiện được lỗi phần mềm. Mặt khác, kỹ thuật “Hộp trắng” lại đòi hỏi hiểu biết về cấu trúc logic bên trong, các cấu trúc điều khiển và các luồng dữ liệu của chương trình. Hai kỹ thuật này không thay thế được nhau mà bổ sung cho nhau.

1. **Vận hành và bảo trì hệ thống**

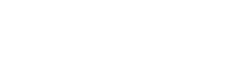
Giai đoạn cuối cùng là cài đặt hệ thống phần mềm trong môi trường sử dụng của khách hàng sau khi sản phẩm đã được kiểm định. Hệ thống sẽ hoạt động, cung cấp các thông tin, xử lý các yêu cầu và thực hiện những gì đã được thiết kế.

Tuy nhiên, vấn đề bảo trì phần mềm hoàn toán khác với bảo trì của phần cứng

# **PHẦN 2: CÁC BƯỚC XÂY DỰNG MÔ HÌNH HÓA USE CASE**

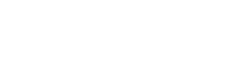
## CHƯƠNG 1: XÁC ĐỊNH CÁC TÁC NHÂN HỆ THỐNG:

1. **Tác nhân (actor):**
   * + Tác nhân (Actor) là một đối tượng bên ngoài hệ thống giao tiếp với hệ thống, thông qua:
       - Tương tác, trao đổi thông tin với hệ thống hoặc sử dụng chức năng của hệ thống.
       - Cung cấp đầu vào hoặc nhận đầu ra của hệ thống.
       - Không điều khiển hoạt động của hệ thống.
     + Ký hiệu:





|  |
| --- |
| <<actor>> Tên tác nhân |
|  |
|  |



<Tên tác nhân>

* + - Tên tác nhân: Là danh từ, theo vài trò của tác nhân.

1. **Phân loại tác nhân:**
   * + Phân loại Tác nhân (actor):
       - Tác nhân chính: Là những người sử dụng những chức năng chính của hệ thống.
       - Tác nhân phụ: Là những nười làm những công việc quản lý, bảo dưỡng hệ thống.
       - Các thiết bị ngoài: Là thiết bị được hệ thống điều khiển.
       - Các hệ thống khác: Là các hệ thống không thuộc hệ thống đang xét nhưng tương tác với nó.
2. **Xác định tác nhân:**

Xác định tác nhân cũng được xem có tầm quan trọng như xác định class, use case, liên kết  
Khi xác định người sử dụng phần mềm hệ thống, chúng ta đừng quan trọng vấn đề quan sát người nào đang sử dụng hệ thống mà chúng ta nên xác định xem vai trò chịu trách nhiệm trong việc sử dụng hệ thống. Nghĩa là tác động lên hệ thống theo nghĩa cung cấp thông tin cho hệ thống hoặc nhận kết quả xử lý từ hệ thống.  
Tác nhân được hiểu là một vai trò tham gia vào hệ thống không giống như một con người cụ hể hoặc một công việc. Một đối tượng có thể tham gia vào một hoặc nhiều vai trò

\* Để tìm các tác nhân có thể trả lời các câu hỏi sau:

- Ai hay hệ thống nào sẽ là người sử dụng những chức năng chính của hệ thống?

- Ai cần sự hỗ trợ của hệ thống để thực hiện những công việc hàng ngày của họ?

- Ai sẽ cần bảo trì, quản trị và đảm bảo cho hệ thống hoạt động?

- Hệ thống sẽ phải xử lý và làm việc với những trang thiết bị phần cứng nào?

- Hệ thống cần phải tương tác với các hệ thống nào khác?

- Ai hay cái gì quan tâm đến kết quả mà hệ thống sẽ sản sinh ra?

## CHƯƠNG 2: XÁC ĐỊNH USE CASE VÀ CÁC MỐI QUAN HỆ:

1. **Use case:**
   * + Một use case là biểu diễn một tập hợp chuỗi các hành động, mà hệ thống thực hiện nhằm cung cấp một kết quả cụ thể cho một tác nhân.
     + Một Use case được xem như là thể hiện chức năng của hệ thống.
     + Một Use case được xem như một chức năng hệ thống từ quan điểm người dùng, như vậy tập hợp tất cả use case biểu diễn bộ mặt của hệ thống bao gồm các chức năng cần có để cung cấp cho các đối tượng tương tác làm việc với hệ thống. Như vậy, use case dùng để mô tả yêu cầu của hệ thống mới về mặt chức năng, mỗi chức năng sẽ được biểu diễn như một hoặc nhiều use case.
     + Ký hiệu:



* + - Đặt tên use case: phải phản ánh mô tả tổng quan về chức năng của use case. Tên gồm: Động từ (do)+ Bổ ngữ (what).



1. **Xác định Use case:**

- Chúng ta bắt đầu từ tập các tác nhân đã xác định trong bước đầu tiên. Ứng với mỗi tác nhân:

- Tìm các nhiệm vụ và chức năng mà tác nhân sẽ thi hành hoặc hệ thống cần tác nhân để thi hành và mô hình hoá nó như là use case. Use case sẽ đại diện một dòng sự kiện dẫn tới một mục tiêu rõ ràng (hoặc trong một vài trường hợp, dẫn tới một vài mục tiêu  
riêng biệt có thể là các phương án thay thế cho tác nhân hoặc cho hệ thống so với  
dòng sự kiện chính)

- Đặt tên cho use case: tên use case nên đặt nhằm phản ánh một mô tả tổng quan về chức năng của use case. Tên nên dễn đạt những gì xảy ra khi một thể hiện của use  
case được thi hành. Một hình thức đặt tên use case phổ biến là : động từ (do) + danh  
từ (what)

- Bắt đầu từ tập tác nhân. Ứng với mỗi tác nhân tìm các nhiệm vụ và chức năng mà tác nhân sẽ thi hành hoặc hệ thống cần tác nhân để thi hành và mô hình hóa nó như là một use case.

- Mô tả use case một cách ngắn gọn bằng việc áp dụng các thuật ngữ gần gũi với người sử dụng. Điều này sẽ làm cho mô tả use case ít mơ hồ.

- Use case sẽ đại diện cho một dòng sự kiện dẫn tới một mục tiêu rõ ràng.

\* Để xác định các Use case có thể đặt các câu hỏi sau:

- Tác nhân đó cần chức năng nào từ hệ thống. Hành động chính của tác nhân này là gì?

- Tác nhân cần phải xem, cập nhật hay lưu trữ thông tin gì trong hệ thống?

- Tác nhân có cần thông báo cho hệ thống những sự kiện nào đó hay không? Những sự kiện như thế đại diện cho những chức năng nào?

- Hệ thống có cần thông báo cho tác nhân khi có thay đổi trong hệ thống hay không?

- Hệ thống cần có những chức năng gì để đơn giản hóa các công việc của tác nhân?

1. **Xác định các mối quan hệ:**

* Quan hệ tác nhân – use case: quan hệ này cho biết tác nhân sẽ tương tác với use case. Một use case luôn luôn được khởi tạo bởi một tác nhân và có thể tương tác với nhiều tác nhân



* Ví dụ:



* Việc mô tả use case có thể sẽ khó hiểu nếu use case này chứa đựng nhiều dòng phụ hoặc dòng ngoại lệ chỉ xử lý cho những sự kiện trong những điều kiện đặc biệt. Để làm đơn giản mô tả này chúng ta sử dụng thêm các mối kết hợp <<extend>> và <<include>>.
  1. **Quan hệ kết hợp**

**-** Là loại quan hệ giữa tác nhân và use case

- Mũi tên cho biết ai là người khởi xưởng giao tiếp, và thể hiện sự trao đổi một chiều.

\* Tác nhân khởi phát usecase sử dụng:





\* Tác nhân và usecase sử dụng trao đổi thông tin với nhau. và Tác nhân nhận kết quả (giá trị) từ usecase sử dụng:



* 1. **Quan hệ gộp**

-*Liên kết sử dụng (<<include>>):* được thành lập khi chúng ta có các use case mà tìm thấy một vài use case có những dòng hoạt động chung, và để tránh mô tả dòng hoạt động chung đó lặp lại trên những use case này, chúng ta có thể tách những dòng hoạt động chung đó ra thành một use case. Use case mới này có thể sử dụng bởi những use case khác. Quan hệ giữa những use case với use case được trích ra này gọi là quan hệ <<include>>. Quan hệ sử dụng giúp chúng ta tránh sự trùng lắp bằng cách cho phép một use case có thể được chia sẽ.

**-** Thể hiện một use case luôn luôn sử dụng chức năng của use case khác. use case sử dụng ở gốc mũi tên là use case sử dụng cơ sở: nó sát nhập một các tường minh (bắt buộc) hành vi của use case sử dụng kia tại vị trí được chỉ rõ trong đặc tả của nó.

- Sử dụng để mô hình hóa một vài chức năng dùng chung, sử dụng lại giữa hai hay nhiều use case.



* 1. **Quan hệ mở rộng**

- *Liên kết mở rộng (<<extend>>):* được dùng khi chúng ta có một use case tương tự như use-case khác nhưng có nhiều hơn một vài xử lý đặc biệt. Giống như liên kết tổng quát – chuyên biệt, trong đó, use case chuyên biệt là một mở rộng của use case tổng quát bằng việc đưa thêm vào các hoạt động hoặc ngữ nghĩa mới vào use case tổng quát, hoặc bỏ qua hoạt động của use case tổng quát.

**-** Một use case tùy ý mở rộng chức năng do use case khác cung cấp

- Sử dụng để chỉ ra rằng một use case được mở rộng từ một use case khác bằng cách thêm vào một chức năng cụ thể. Use case sử dụng ở đầu mũi tên là Use case sử dụng cơ sở: nó sát nhập một cách không tường minh (tùy theo những điều kiện nào đó) hành vi của ca sử dụng kia tại một số điểm đặc biệt trong kịch bản của nó.

- Sử dụng để mô hình hóa một vài chức năng dùng chung, sử dụng lại giữa hai hay nhiều use case.

- Dùng quan hệ mở rộng để chỉ rõ:

+ Hành vi tùy chọn.

+ Hành vi chỉ xảy ra dưới một số điều kiện nào đó.

+ Nhiều luồng sự kiện có thể xảy ra theo sự lựa chọn của tác nhân.



* 1. **Quan hệ khái quát**

**-** Chỉ ra một vài tác nhân hay use case có một số cái chung, giống nhau.

- Không nhất thiết hình thành quan hệ này cho các tác nhân khi:

+ Khi một loại tác nhân kích hoạt một hay vài use case mà loại tác nhân khác không kích hoạt -> nên hình thành quan hệ khái quát hóa.

+ Khi cả hai loại tác nhân cùng sử dụng các use case -> không cần mô hình hóa quan hệ khái quát hóa .



1. **Xây dựng biểu đồ Use case:**

Các chú ý khi xây dựng biểu đồ Use case:

- Không nên mô hình hóa quan hệ kết hợp giữa tác nhân với tác nhân -> vì giao tiếp giữa các tác nhân là ở bên ngoài hệ thống.

-> Hãy sử dụng biểu đồ luồng công việc để khảo sát quan hệ giữa các tác nhân.

- Không hình thành quan hệ kết hợp (Association) giữa các use case.

-> Biểu đồ chỉ có thể chỉ ra có các use case nào nhưng không chỉ ra trật tự thực hiện chúng.

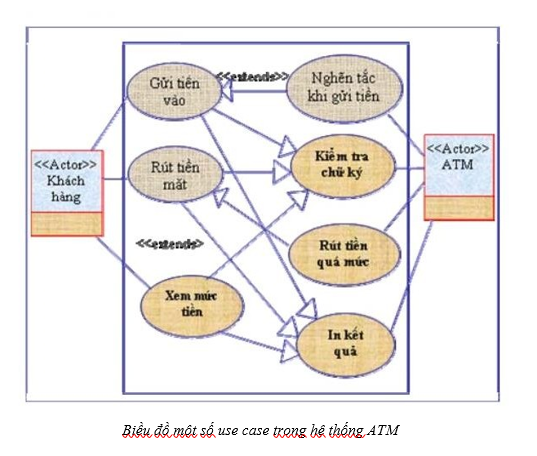
- Mỗi use case phải có tác nhân kích hoạt (trừ use case trong quan hệ extends và quan hệ includes).

-> Nên vẽ mũi tên thể hiện association đi từ tác nhân đến use case.

- Có thể xem CSDL là lớp ở dưới biểu đồ use case

-> Có thể nhập tin vào CSDL ở use case này và xâm nhập dữ liệu trong CSDL ở use case khác.

-> Không vẽ association giữa các use case để chỉ ra luồng thông tin.



1. **Phân chia biểu đồ Use case thành các gói:**

- Mỗi use case minh họa một kịch bản trong hệ thống.

- Hệ thống phức tạp nên phân chia thành các gói để dễ quan sát và quản lí.

- Mỗi gói phản ánh một phạm vi của hệ thống.

1. **Đặc tả biểu đồ Use case:**

Ví dụ: Đặt tả use-case tìm kiếm sách

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên use case | Tra cứu Sách |
| Tác nhân chính: | Người dùng |
| Người chịu trách nhiệm: | Người dùng |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu. |
| Đảm bảo tối thiểu: | Đăng nhập thành công vào phần mềm. |
| Đảm bảo thành công: | Tra cứu thành công Sách |
| Kích hoạt: | Người dùng đăng nhập vào phần mềm, chọn tab <Quản Lý Sách> sau đó nhấn vào biểu tượng  , người dùng nhập các trường cần tra cứu vào sau đó nhấn tìm kiếm |
| Chuỗi sự kiện chính: | 1. Hệ thống hiển thị form tra cứu Sách. 2. Người dùng nhấp Hiển thị. 3. Hệ thống hiển thị danh sách Sách tương ứng với chức năng đã chọn. |
| Ngoại lệ: |  |

# **PHẦN 3: MÔ HÌNH HÓA PHẦN MỀM “QUẢN LÝ NHÀ SÁCH”**

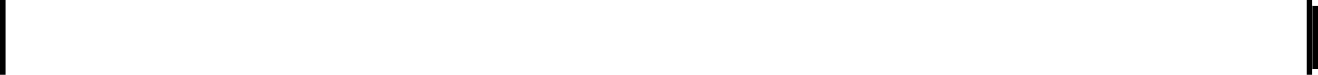
## CHƯƠNG 1: MÔ TẢ NGHIỆP VỤ

1. **Danh sách các yêu cầu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên yêu cầu** | **Biểu mẫu** | **Quy định** | **Ghi chú** |
| 1 | Lập phiếu nhập sách | BM1 | QĐ1 |  |
| 2 | Lập hóa đơn bán sách | BM2 | QĐ2 |  |
| 3 | Tra cứu sách | BM3 |  |  |
| 4 | Lập phiếu thu tiền | BM4 | QĐ4 |  |
| 5 | Lập báo cáo tháng | BM5 |  |  |
| 6 | Thay đổi quy định |  | QĐ6 |  |

1. **Danh sách các biểu mẫu và quy định**
   1. **Biểu mẫu 1 và quy định 1**

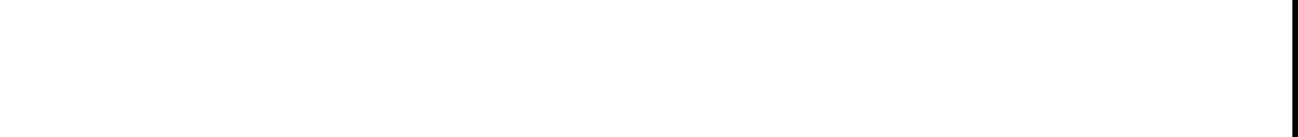
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BM1** | | **Phiếu nhập sách** | | | |
| Ngày nhập sách: ................. | | | | | |
| **STT** | **Sách** | | **Thể loại** | **Tác giả** | **Số lượng** |
| 1 |  | |  |  |  |
| 2 |  | |  |  |  |



**QĐ1: Số lượng nhập ít nhất là 150. Chỉ nhập các đầu sách có lượng tồn ít hơn 300.**

* 1. **Biểu mẫu 2 và quy định 2**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **BM2** | | **Hóa Đơn Bán Sách** | | | | |
| Họ tên khách hàng: ................ | | | | Ngày lập hóa đơn: .................. | | |
| **STT** | **Sách** | | **Thể loại** | | **Số lượng** | **Đơn giá** |
| 1 |  | |  | |  |  |
| 2 |  | |  | |  |  |



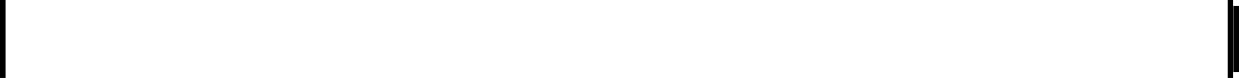
**QĐ2: Chỉ bán cho các khách hàng nợ không quá 20.000 và đầu sách có lượng tồn sau khi bán ít nhất là 20.**

* 1. **Biểu mẫu 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BM3** | | **Danh sách sách** | | | |
| **STT** | **Tên sách** | | **Thể loại** | **Tác giả** | **Số lượng** |
| 1 |  | |  |  |  |
| 2 |  | |  |  |  |

* 1. **Biểu mẫu 4 và quy định 4**

|  |  |  |
| --- | --- | --- |
| **BM4** | **Phiếu thu tiền** | |
| Họ tên khách hàng: ................... | | Địa chỉ:...................................... |
| Điện thoại: .................................. | | Email: ........................................ |
| Ngày thu tiền: ............................ | | Số tiền thu: ............................... |



**QĐ4: Số tiền thu không vượt quá số tiền khách hàng đang nợ**

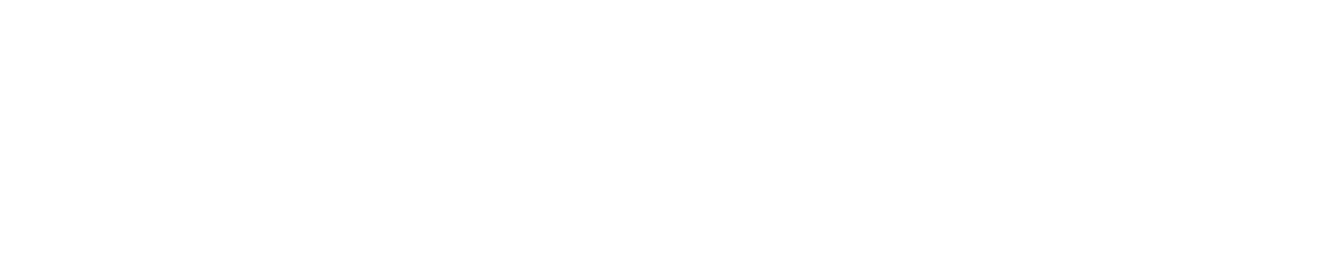
* 1. **Biểu mẫu 5**
     1. **Biểu mẫu 5 .1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BM5.1** | | **Báo cáo tồn** | | | |
| Tháng: ……………… | | | | | |
| **STT** | **Sách** | | **Tồn đầu** | **Phát sinh** | **Tồn cuối** |
| 1 |  | |  |  |  |
| 2 |  | |  |  |  |

* + 1. **Biểu mẫu 5 .2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **BM5.2** | | **Báo cáo công nợ** | | | |
| Tháng: ………………… | | | | | |
| **STT** | **Khách hàng** | | **Nợ đầu** | **Phát sinh** | **Nợ cuối** |
| 1 |  | |  |  |  |
| 2 |  | |  |  |  |

* 1. **Quy định 6**



**QĐ6: Người dùng có thể thay đổi các qui định như sau:**

**+ QĐ1: Thay đổi số lượng nhập tối thiểu, lượng tồn tối đa trước khi nhập.**

**+ QĐ2: Thay đổi tiền nợ tối đa, lượng tồn tối thiểu sau khi bán.**

**+ QĐ4: Sử dụng hay không sử dụng qui định này.**

1. **Bảng trách nhiệm yêu cầu nghiệp vụ**

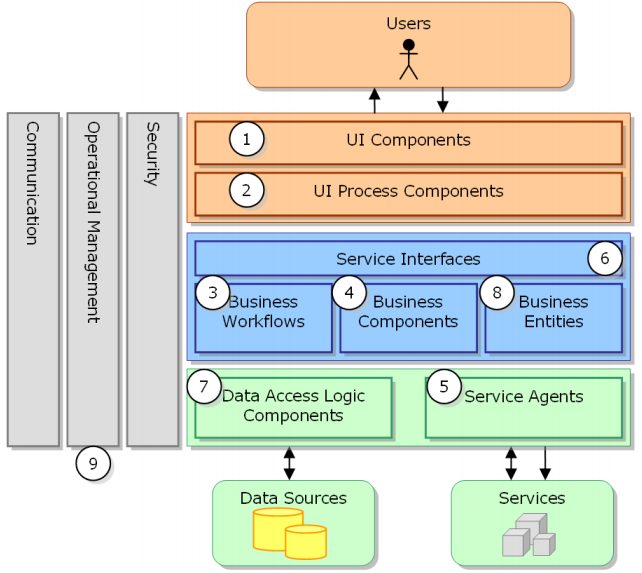
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nghiệp vụ** | **Người dùng** | **Phần mềm** | **Ghi chú** |
| 1 | Lập phiếu nhập sách | * Cung cấp thông tin về ngày nhập * Danh sách các đầu sách nhập cùng với thông tin chi tiết liên quan (Tên sách, Thể loại, Tác giả, Số   lượng nhập) | - Kiếm tra quy định, lưu thông tin về phiếu nhập nhập sách | - Cho phép xem trước, hủy và cập nhật lại thông tin trên phiếu trước khi lưu |
| 2 | Lập hóa đơn bán sách | * Cung cấp thông tin về Họ tên khách hàng, ngày lập hóa đơn * Danh sách các đầu sách bán cùng với thông tin chi tiết liên quan (Tên sách, Thể loại, Số lượng bán, | - Kiểm tra quy định, lưu thông tin hóa đơn bán sách | - Cho phép xem trước, hủy, cập nhật lại thông tin trước khi lưu |
| 3 | Tra cứu sách | Cung cấp ít nhất một  trong các thông tin về  đầu sách cần tìm  - Thông tin trực tiếp: Tên sách, thể loại, tác  giả, số lượng tồn (từ…  đến…)  - Thông tin gián tiếp: Số  lượng bán (từ … đến …),  Tổng số lượng bán (từ …  đến …), Ngày bán (từ …  đến …), Đơn giá bán (từ  … đến …) | - Tìm kiếm và trả lại  thông tin về  các đầu sách thỏa tiêu chuẩn tìm kiếm |  |
| 4 | Lập phiếu thu  tiền | - Cung cấp thông tin về  khách hàng (Họ tên  khách hàng, Địa chỉ,  Điện thoại, Email),  Ngày thu tiền, Số tiền  Thu | -Kiểm tra quy định  và in ra phiếu thu  tiền | - Cho phép  xem trước,  hủy và cập  nhật lại thông  tin trên phiếu  trước khi in |
| 5 | Lập báo cáo tồn | - Cho biết tháng lập báo cáo tồn | - In ra báo cáo tồn |  |
| 6 | Lập báo cáo công nợ | - Cho biết tháng lập báo cáo công nợ | - In ra báo cáo công nợ |  |
| 7 | Thay đổi quy định | - Cung cấp thông tin về danh sách các quy định cần thay đổi với các thông tin chi tiết liên quan (số lượng nhập tối thiểu, số lượng tồn tối đa trước khi nhập, tiền nợ tối đa, lượng tồn tối thiểu sau khi bán) | - Ghi nhận lại quy định mới |  |

## CHƯƠNG 2: KIẾN TRÚC HỆ THỐNG

1. **Kiến trúc hệ thống**

#### Mô hình hệ thống

* Hệ thống sử dụng mô hình 3 lớp (3tier)



|  |  |  |
| --- | --- | --- |
| **STT** | **Thành phần** | **Giải thích** |
| 1 | Presenration (UI Component,UI Process Component) | Đây là layer tạo lên giao diện cho người dùng, là nơi tiếp nhận và kết xuất ra kết quả của chương trình  Nó có nhiệm vụ xử lý, kiểm tra các dữ liệu nhập vào (Ví dụ như kiểm tra nội dung của Textbox)  Tiếp nhận các event của người dùng, kiểm tra dữ liệu được nhập vào, gửi yêu cầu xử lý xuống tầng kế tiếp. |
| 2 | BUS | Thành phần xử lý logic theo đúng nghiệp vụ  Xử lý logic và validation |
| 3 | DAL | Thành phần làm việc trực tiếp với cơ sở dữ liệu |
| 4 | DTO | Đây là gói dữ liệu, thành phần trung gian đại diện cho một mô hình dữ liệu để trao đổi giữa các lớp. |

* 1. **Cách thức hoạt động**
* Presentation: Còn gọi là GUI, là giao diện mà người dùng có thể nhìn thấy. Tại đây tiếp nhận và xử lý mọi hành vi của người dùng. Khi sự kiện (event) xảy ra, nếu đã được kiểm duyệt (validation) thì sẽ gửi dữ liệu xuống tầng BUS để tiếp tục xử lý nghiệp vụ
* BUS: Tầng này sẽ phụ trách việc xử lý logic nghiệp vụ. Mọi thông tin sẽ được xử lý ở tầng này, sau đó hoặc cập nhật giao diện cho tầng GUI hoặc xử lý với cơ sở dữ liệu thông qua DAL.
* DAL: Tầng này phụ trách việc xử lý trực tiếp với cơ sở dữ liệu. Mọi phương thức thêm, cập nhật, xoá, sửa (CRUD) hoặc các phương thức khác đều được thực hiện ở đây. Trong ứng dụng này, tất cả mọi phương thức đều được thực hiện thông qua procedure (SQL server).
* DTO: Đây không phải là một tầng trong Sơ đồ 3 lớp: Đây là gói dữ liệu, được viết dưới dạng lớp (Class), để truyền tải dữ liệu dữ các tầng, thường được gọi là Model. Hay nói cách khác, lớp này chính là đại diện cho các lớp của Cơ sở dữ liệu.

1. **Sơ đồ Use Case tổng quát**



* 1. **Danh sách Actor**

|  |  |  |
| --- | --- | --- |
| **STT** | **Actor** | **Mô tả** |
| **1** | Quản lý | Là người có quyền cao nhất, có thể sử dụng mọi chức năng của phần mềm. |
| **2** | Thủ thư | Là người sử dụng phần mềm bị giới hạn một số chức năng: Thay đổi quy định, thêm người dùng và sửa thông tin người dùng |

* 1. **Danh sách Use – Case**

|  |  |
| --- | --- |
| **STT** | **Tên Use case** |
| 1 | Đăng nhập |
| 2 | Đăng xuất |
| 3 | Thêm sách |
| 4 | Sửa thông tin sách |
| 5 | Tra cứu sách |
| 6 | Thêm khách hàng |
| 7 | Sửa thông tin khách hàng |
| 8 | Tra cứu khách hàng |
| 9 | Lập phiếu nhập sách |
| 10 | Xuất phiếu nhập sách |
| 11 | Lập phiếu thu tiền |
| 12 | Xuất phiếu thu tiền |
| 13 | Lập hóa đơn bán sách |
| 14 | Xuất hóa đơn bán sách |
| 15 | Lập báo cáo tồn |
| 16 | Xuất báo cáo tồn |
| 17 | Lập báo cáo nợ công |
| 18 | Xuất báo cáo nợ công |
| 19 | Thêm người dùng |
| 20 | Sửa thông tin người dùng |
| 21 | Thay đổi quy định |
| 22 | Đổi mật khẩu |

## CHƯƠNG 3: XÂY DỰNG BIỂU ĐỒ UML

1. Usecase Xem thông tin hội thảo
2. Usecase diagram

A close up of a map

Description automatically generated

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên usecase: | Xem thông tin hội thảo |
| Tác nhân chính: | Người tham gia |
| Người chịu trách nhiệm: | Người tham gia |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu |
| Đảm bảo tối thiểu: | N/A |
| Đảm bảo thành công: | Xem thông tin hội thảo thành công |
| Kích hoạt: |  |
| Chuỗi sự kiện chính: | 1. Người dùng chọn hội thảo cần xem thông tin  2. Hệ thống hiển thị thông tin hội thảo |
| Ngoại lệ: |  |

1. Activity diagram

A screenshot of a cell phone

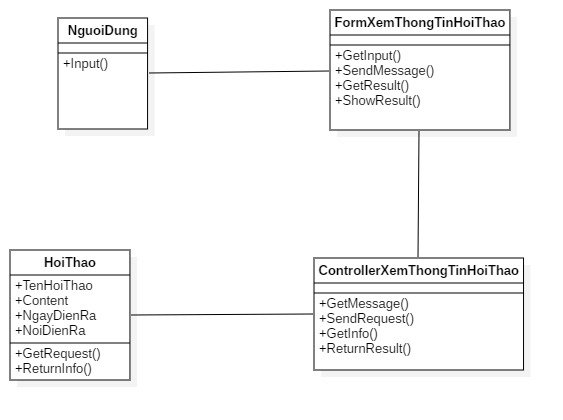
Description automatically generated

1. Sequence diagram

A screenshot of a social media post

Description automatically generated

1. Class diagram



1. Usecase Đăng ký
2. Usecase diagram

A close up of text on a white background

Description automatically generated

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên usecase: | Đăng ký |
| Tác nhân chính: | Người tham gia |
| Người chịu trách nhiệm: | Người tham gia |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu |
| Đảm bảo tối thiểu: | N/A |
| Đảm bảo thành công: | Đăng ký thành công |
| Kích hoạt: |  |
| Chuỗi sự kiện chính: | 1. Người dùng nhập thông tin tài khoản.  2. Hệ thống kiểm tra  3. Hệ thống thông báo đăng ký thành công |
| Ngoại lệ: |  |

1. Activity diagram

A close up of a piece of paper

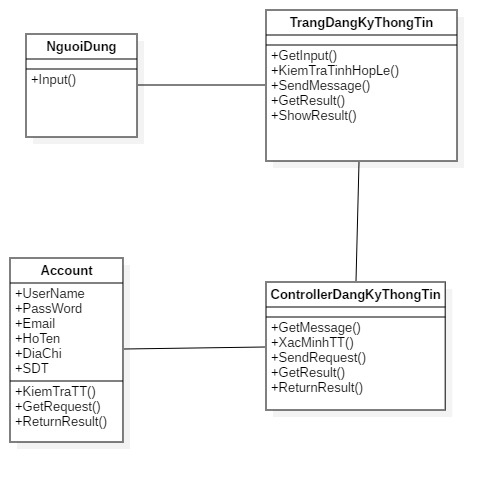
Description automatically generated

1. Sequence diagram

A screenshot of a social media post

Description automatically generated

1. Class diagram



1. Usecase Xem thông tin vé đã đặt
2. Usecase diagram

A close up of text on a white background

Description automatically generated

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên usecase: | Xem thông tin vé đã đặt |
| Tác nhân chính: | Người tham gia |
| Người chịu trách nhiệm: | Người tham gia |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu |
| Đảm bảo tối thiểu: | Người tham gia đăng nhập thành công vào hệ thống |
| Đảm bảo thành công: | Hiển thị thông tin vé đã đặt |
| Kích hoạt: |  |
| Chuỗi sự kiện chính: | 1. Người tham gia đăng nhập vào hệ thống.  2. Hệ thống kiểm tra  3. Hệ thống hiển thị thông tin vé đã đặt |
| Ngoại lệ: |  |

1. Activity diagram

A picture containing screenshot

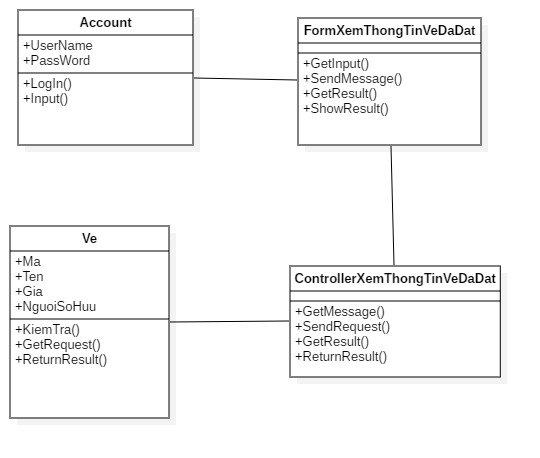
Description automatically generated

1. Sequence diagram

A screenshot of a cell phone

Description automatically generated

1. Class diagram



1. Usecase Gửi phản hồi
2. Usecase diagram

A close up of text on a black background

Description automatically generated

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên usecase: | Gửi phản hồi |
| Tác nhân chính: | Người tham gia |
| Người chịu trách nhiệm: | Người tham gia |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu |
| Đảm bảo tối thiểu: | N/A |
| Đảm bảo thành công: | Gửi phản hồi thành công |
| Kích hoạt: |  |
| Chuỗi sự kiện chính: | 1. Người tham gia gửi phản hồi  2. Hệ thống kiểm tra và lưu thông tin phản hồi  3. Hệ thống thông báo gửi phản hồi thành công |
| Ngoại lệ: |  |

1. Activity diagram

A screenshot of a cell phone

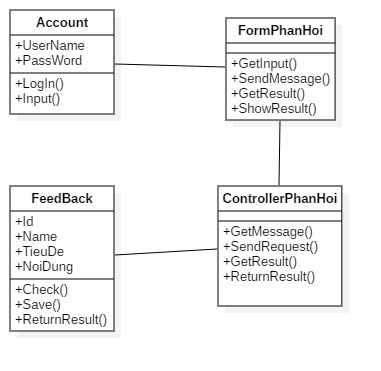
Description automatically generated

1. Sequence diagram

A screenshot of a cell phone

Description automatically generated

1. Class diagram



1. Usecase Gửi thắc mắc/ yêu cầu
2. Usecase diagram

A close up of text on a white background

Description automatically generated

|  |  |
| --- | --- |
| ***Đặc tả*** | ***Ý nghĩa*** |
| Tên usecase: | Gửi thắc mắc/ yêu cầu |
| Tác nhân chính: | Người tham gia |
| Người chịu trách nhiệm: | Người tham gia |
| Tiền điều kiện: | Kết nối cơ sở dữ liệu |
| Đảm bảo tối thiểu: | Tài khoản của người tham gia đã tồn tại |
| Đảm bảo thành công: | Giải đáp về thắc mắc/ yêu cầu sẽ được gửi đến người tham gia bằng email |
| Kích hoạt: |  |
| Chuỗi sự kiện chính: | 1. Người gửi thắc mắc/ yêu cầu  2. Hệ thống kiểm tra  3. Hệ thống gửi giải đáp thắc mắc/ yêu cầu tới người dùng |
| Ngoại lệ: |  |

1. Activity diagram

A picture containing screenshot

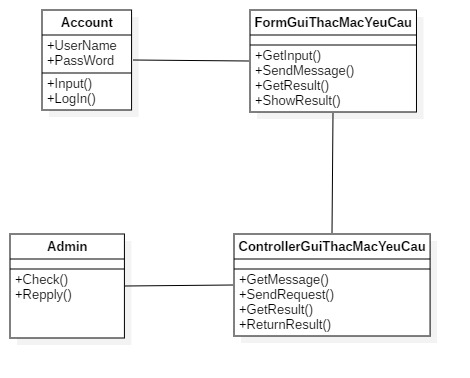
Description automatically generated

1. Sequence diagram

A screenshot of a cell phone

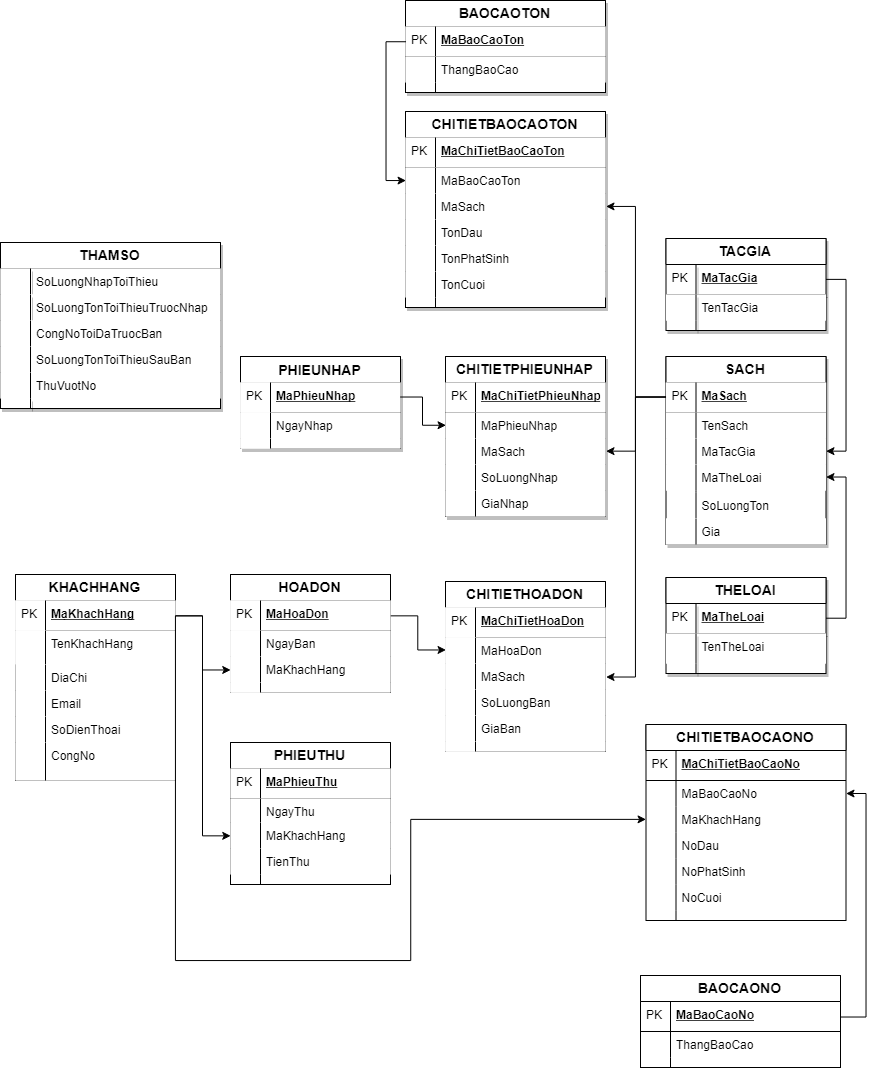
Description automatically generated

1. Class diagram



## CHƯƠNG 3: THIẾT KẾ DỮ LIỆU

1. **Sơ đồ**



1. **Diễn giải**
   1. **Bảng PHIEUNHAP**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaPhieuNhap** | Short Text | Mã của phiếu nhập |
| 2 | NgayNhap | Date/Time | Ngày nhâp sách |

* 1. **Bảng CHITIETPHIEUNHAP**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaChiTietPhieuNhap** | Short Text | Mã của mỗi chi tiết phiếu nhập |
| 2 | MaPhieuNhap | Short Text | Mã của phiếu nhập |
| 3 | MaSach | Short Text | Mã của đầu sách |
| 4 | SoLuongNhap | Integer | Số lượng nhập của đầu sách |

* 1. **Bảng SACH**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaSach** | Short Text | Mã của đầu sách |
| 2 | TenSach | Short Text | Tên của đầu sách |
| 3 | TacGia | Short Text | Tác giả của đầu sách |
| 4 | TheLoai | Short Text | Thể loại của đầu sách |
| 5 | SoLuongTon | Integer | Số lượng tồn của đầu sách |

* 1. **Bảng CHITIETHOADON**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaChiTietHoaDon** | Short Text | Mã của chi tiết hóa đơn |
| 2 | MaHoaDon | Short Text | Mã của hóa đơn bán sách |
| 3 | MaSach | Short Text | Mã của đầu sách trong hóa đơn |
| 4 | DonGiaBan | Short Text | Đơn giá bán của mỗi đầu sách trong hóa đơn |
| 5 | SoLuongBan | Integer | Số lượng tồn của mỗi đầu sách trong hóa đơn |

* 1. **Bảng HOADON**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaHoaDon** | Short Text | Mã của hóa đơn |
| 2 | MaKhachHang | Short Text | Mã của khách hàng |
| 3 | NgayLapHoaDon | Date/Time | Ngày lập hóa đơn |
| 4 | TongThanhTien | Integer | Tổng thành tiền của hóa đơn |

* 1. **Bảng KHACHHANG**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaKhachHang** | Short Text | Mã của khách hàng |
| 2 | HoTenKhachHang | Short Text | Họ tên của khách hàng |
| 3 | Email | Short Text | Email của khách hàng |
| 4 | DiaChi | Short Text | Địa chỉ của khách hàng |
| 5 | DienThoai | Short Text | Điện thoại của khách hàng |
| 6 | SoTienNo | Integer | Số tiền nợ của khách hàng |

* 1. **Bảng PHIEUTHUTIEN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaPhieuThuTien** | Short Text | Mã của phiếu thu tiền |
| 2 | MaKhachHang | Short Text | Mã của khách hàng cẩn thu tiền |
| 3 | NgayThuTien | Date/Time | Ngày thu tiền |
| 4 | SoTienThu | Integer | Số tiền thu |

* 1. **Bảng BAOCAOTON**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaBaoCaoTon** | Short Text | Mã của báo cáo tồn |
| 2 | ThangBaoCao | Integer | Tháng lập báo cáo tồn |

* 1. **Bảng CHITIETBAOCAOTON**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaChiTietBaoCaoTon** | Short Text | Mã của chi tiết báo cáo tồn |
| 2 | MaBaoCaoTon | Short Text | Mã báo cáo tồn |
| 3 | MaSach | Short Text | Mã sách của các đầu sách trong báo cáo tồn |
| 4 | TonDau | Integer | Số lượng tồn đầu của mỗi đầu sách trong báo cáo tồn |
| 5 | TonPhatSinh | Integer | Số lượng phát sinh của mỗi đầu sách trong báo cáo tồn |
| 6 | TonCuoi | Integer | Số lượng tồn cuối của mỗi đầu sách trong báo cáo tồn |

* 1. **Bảng BAOCAOCONNO**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaBaoCaoCongNo** | Short Text | Mã của báo cáo công nợ |
| 2 | ThangBaoCao | Integer | Tháng lập báo cáo công nợ |

* 1. **Bảng CHITIETBAOCAOCONGNO**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaChiTietBaoCaoCongNo** | Short Text | Mã của chi tiết báo cáo công nợ |
| 2 | MaBaoCaoCongNo | Short Text | Mã báo cáo công nợ |
| 3 | MaKhachHang | Short Text | Mã của khách hàng |
| 4 | NoDau | Integer | Số lượng nợ của mỗi khách hàng  trong báo cáo công nợ |
| 5 | NoPhatSinh | Integer | Số lượng nợ phát sinh của mỗi khách hàng trong báo cáo công nợ |
| 6 | NoCuoi | Integer | Số lượng nợ cuối của mỗi khách hang trong báo cáo công nợ |

* 1. **Bảng THAMSO**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | SoLuongNhapToiThieu | Integer | Số lượng nhập tối thiểu của các đầu  sách trước khi nhập |
| 2 | SoLuongTonToiDa | Integer | Số lượng tồn tối đa của từng đầu sách  trước khi nhập |
| 3 | SoLuongTonToiThieu | Integer | Số lượng tồn tối thiểu của từng đầu  sách sau khi bán |
| 4 | SoTienNoToiDa | Integer | Tiền nợ tối đa của khách hàng |
| 5 | ApDungQuyDinhTienThu KhongVuotTienNo | Yes/No | - Giá trị Yes: áp dụng quy định tiền thu không vượt quá tiền nợ  - Giá trị No: không áp dụng quy định  tiền thu không vượt quá tiền nợ |

* 1. **Bảng LOAINGUOIDUNG**

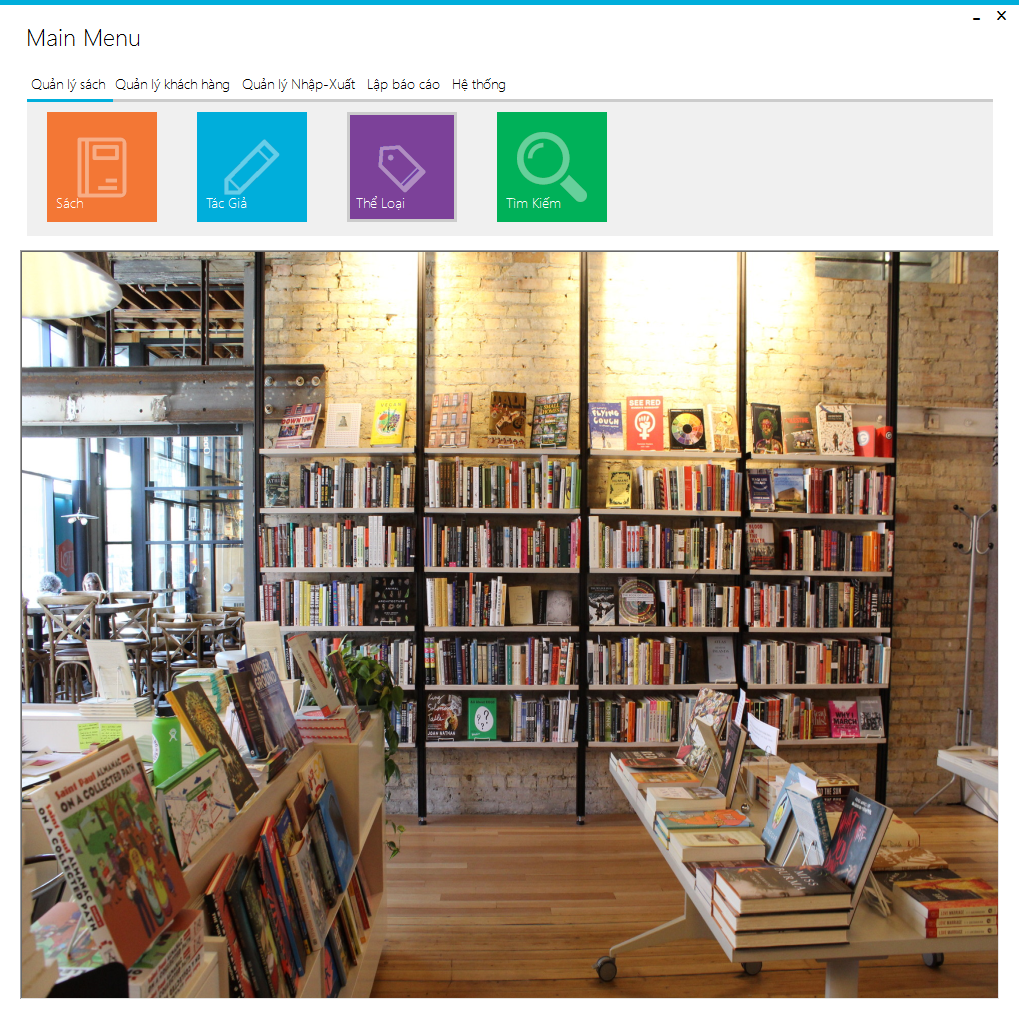
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaLoaiNguoiDung** | Short Text | Mã của loại người dùng |
| 2 | TenLoaiNguoiDung | Short Text | Tên của loại người dùng |

* 1. **Bảng NGUOIDUNG**

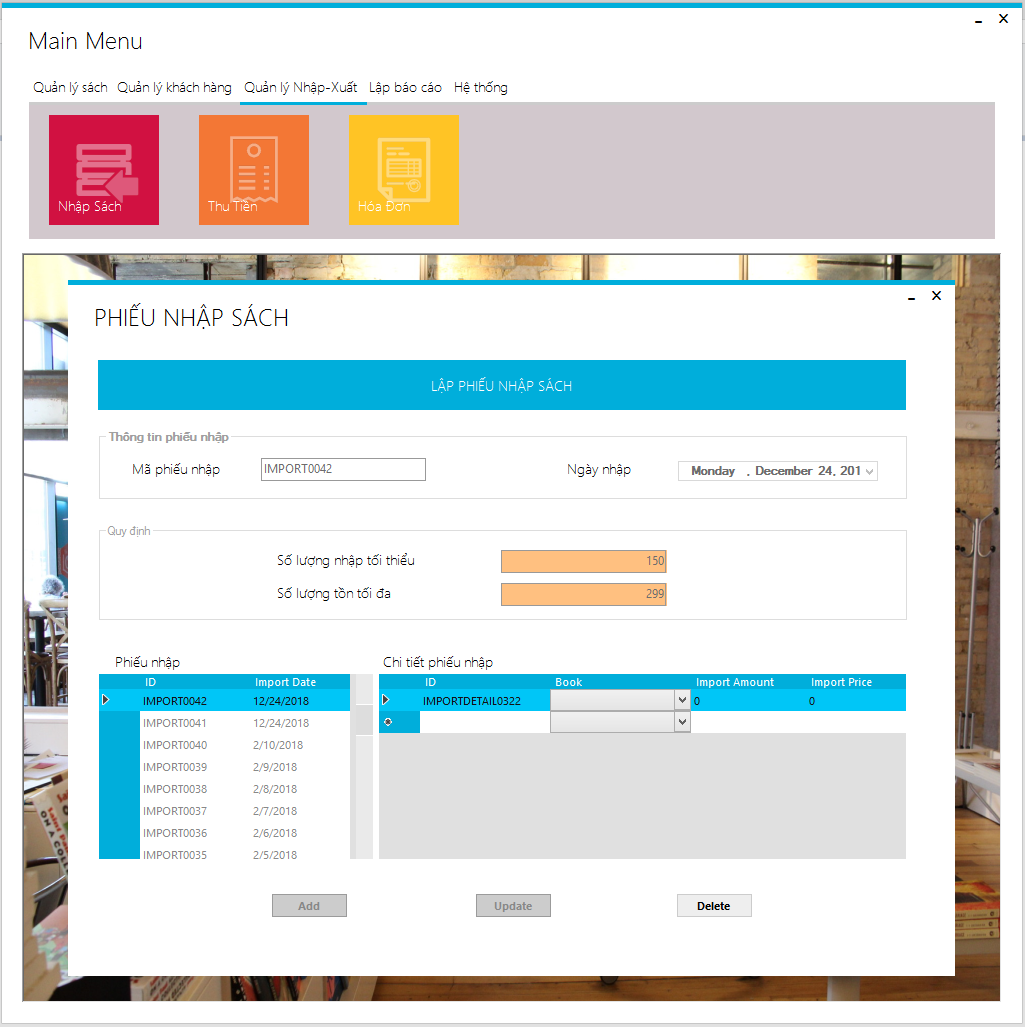
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** |
| 1 | **MaNguoiDung** | Short Text | Mã của người dùng |
| 2 | HoTenNguoiDung | Short Text | Họ tên của người dùng |
| 3 | TenDangNhap | Short Text | Tên đăng nhập |
| 4 | MatKhau | Short Text | Mật khẩu |
| 5 | MaLoaiNguoiDung | Short Text | Mã loại người dùng |

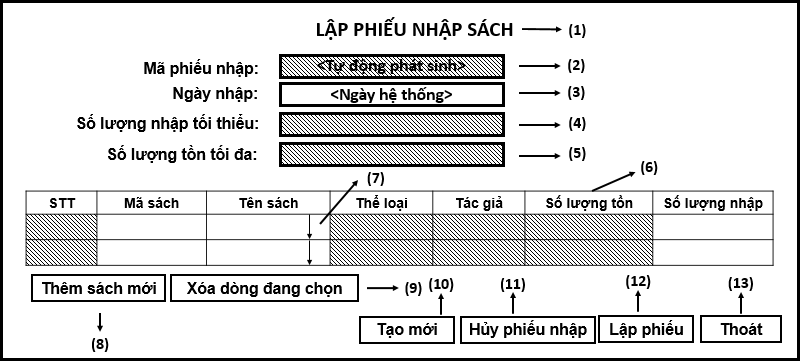
**CHƯƠNG 4: THIẾT KẾ GIAO DIỆN**

### Màng hình chính

****

### Màng hình “Lập phiếu nhập sách”

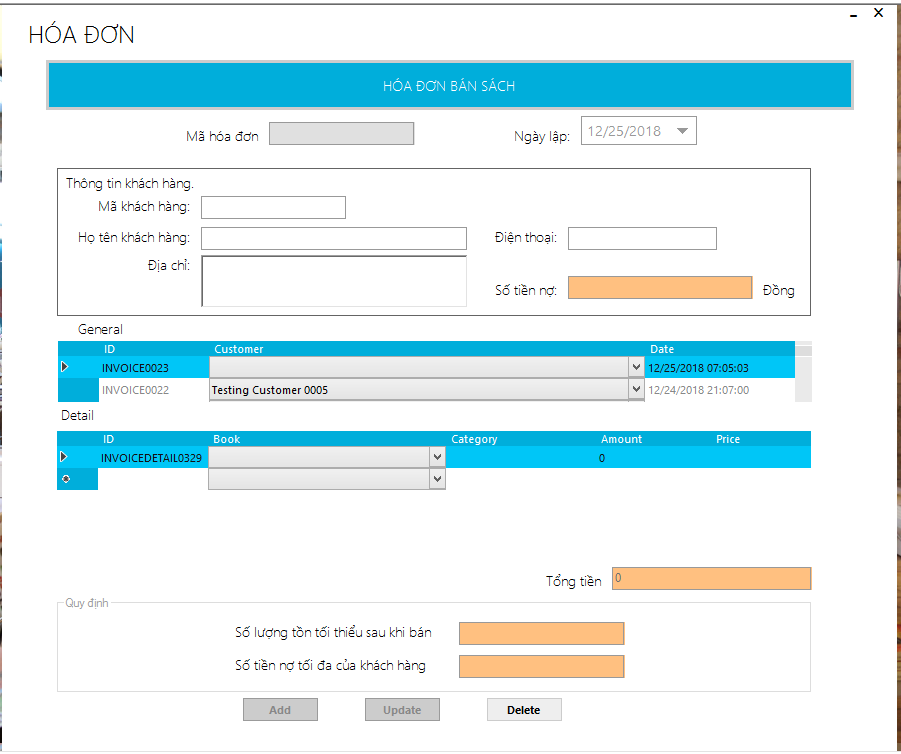
****

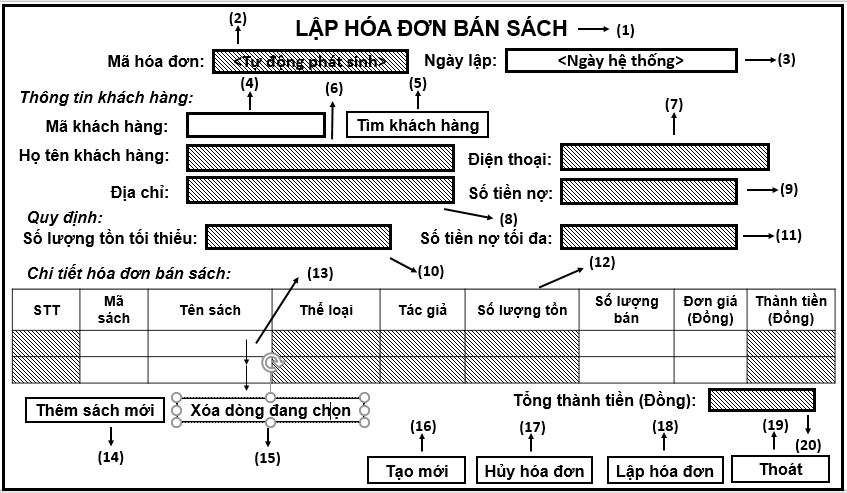


**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblLapPhieuNhapSach | Label | Tiêu đề của màn hình lập phiếu nhập sách |  |
| 2 | txtMaSach | Textbox | Mã sách |  |
| 3 | txtNgayNhap | Textbox | Ngày lập phiếu nhập sách |  |
| 4 | txtSoLuongNhapToiT hieu | Textbox | Số lượng nhập tối thiểu của các đầu sách sách phiếu nhập |  |
| 5 | txtSoLuongTonToiDa | Text | Số lượng tồn tối đa của các đầu sách trong phiếu nhập |  |
| 6 | gridDanhSachSachNhap | GridContro l | Lưới thể hiện thông tin các đầu sách nhập với các thông tin chi tiết liên quan: MaSach, TenSach, TheLoai, TacGia, SoLuongTon, SoLuongNhap |  |
| 7 | cbclmTenSach | Combobox Column | Danh sách tên của các đầu sách |  |
| 8 | btnThemSachMoi | Button | Hiện thị màn hình Tìm kiếm sách để chọn các đầu sách cần  nhập |  |
| 9 | btnXoaDongDangChon | Button | Nút này cho phép xóa dòng đang chọn trên lưới danh sách các đầu sách nhập |  |
| 10 | btnTaoMoi | Button | Nút này cho phép tạo mới phiếu nhập sách |  |
| 11 | btnHuyPhieuNhap | Button | Nút này cho phép khởi tạo lại giá trị mặc định cho cho phiếu nhập |  |
| 12 | btnLapPhieu | Button | Nút này cho phép kiểm tra các quy định lưu lại phiếu nhập sách mới trên cơ sở dữ liệu với các  thông tin về phiếu nhập | **Xử lý chính** |
| 13 | btnThoat | Button | Nút này cho phép thoát màn hình lập phiếu nhập sách |  |

### Màng hình “Lập hóa đơn bán sách”

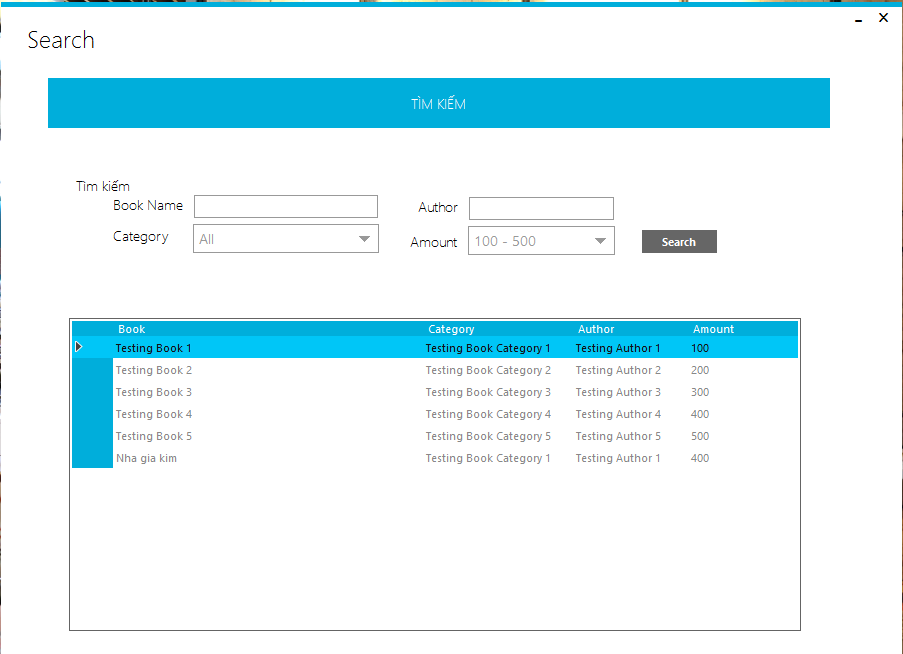
****

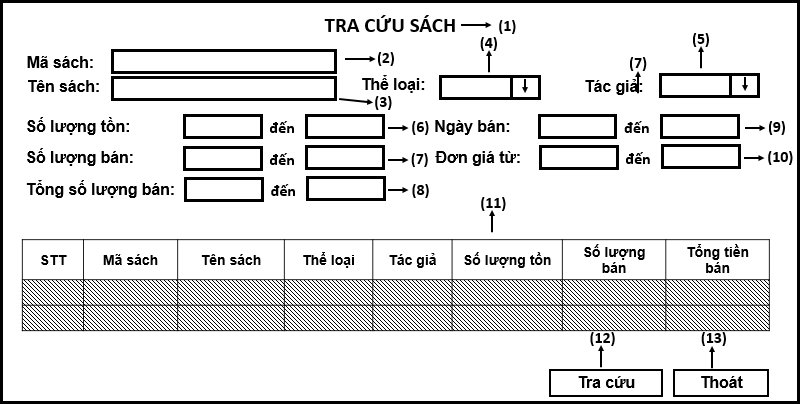


**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblHoaDonBanSach | Label | Tiêu đề của màn hình lập hóa đơn bán sách |  |
| 2 | txtMaHoaDon | Textbox | Mã hóa đơn |  |
| 3 | txtNgayLap | Textbox | Ngày lập của hóa đơn |  |
| 4 | txtMaKhachHang | Textbox | Mã của khách hàng bán |  |
| 5 | btnTimKhachHang | Button | Hiện thị màn hình Tìm kiếm khách hàng để chọn  khách hàng bán |  |
| 6 | btnHoTenKhachHang | Button | Họ tên của khách hàng bán |  |
| 7 | txtDienThoai | Textbox | Điện thoại của khách hàng bán |  |
| 8 | txtDiaChi | Textbox | Địa chỉ của khách hàng bán |  |
| 9 | txtSoTienNo | Textbox | Số tiền khách hàng đang nợ |  |
| 10 | txtSoLuongTonToiThieu | Textbox | Số lượng tồn tối thiểu của các đầu sách sau khi bán |  |
| 11 | txtSoTienNoToiDa | Textbox | Số tiền nợ tối đa của khách hàng để bán |  |
| 12 | gridDanhSachSachBan | GridControl | Lưới thể hiện thông tin các đầu sách bán với các thông tin chi tiết liên quan: MaSach, TenSach, TheLoai, TacGia, SoLuongTon, SoLuongBan, DonGia,  ThanhTien |  |
| 13 | cbclmTenSach | Combobox Column | Danh sách tên của các đầu sách |  |
| 14 | btnThemSachMoi | Button | Hiện thị màn hình Tìm kiếm sách để chọn các đầu sách cần bán |  |
| 16 | btnTaoMoi | Button | Nút này cho phép tạo mới hóa đơn bán sách |  |
| 17 | btnHuyHoaDon | Button | Nút này cho phép khởi  tạo lại giá trị mặc định cho cho hóa đơn bán sách |  |
| 18 | btnLapHoaDon | Button | Nút này cho phép kiểm tra các quy định lưu lại hóa đơn mới trên cơ sở dữ liệu với các thông tin về phiếu nhập | **Xử lý chinh** |
| 19 | btnThoat | Button | Nút này cho phép thoát màn hình lập hóa đơn bán sách |  |
| 20 | txtTongThanhTien | Textbox | Tổng thành tiền của hóa đơn |  |

### Màng hình “tra cứu”

****

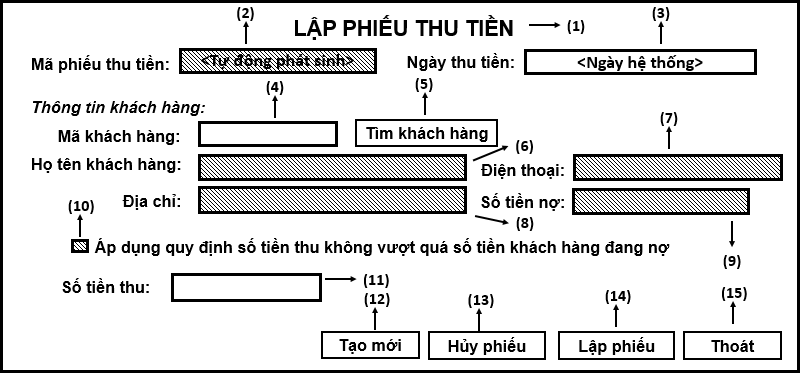


**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblTraCuuSach | Label | Tiêu đề của màn hình tra cứu sách |  |
| 2 | lxtMaSach | Textbox | Tìm kiếm theo mã sách |  |
| 3 | txtTenSach | Textbox | Tìm kiếm theo tên sách |  |
| 4 | cbTheLoai | Combobox | Danh sách các thể loại để tìm kiếm |  |
| 5 | cbTacGia | Combobox | Danh sách các tác giả để tìm kiềm |  |
| 6 | txtSoLuongTonTu  txtSoLuongTonDen | Textbox | Tìm kiếm theo số lượng tồn trong khoảng |  |
| 7 | txtSoLuongBanTu  txtSoLuongBanDen | Textbox | Tìm kiếm theo số lượng bán trong khoảng |  |
| 8 | txtTongSoLuongBanTu  txtTongSoLuongBanDen | Textbox | Tìm kiếm theo tổng số lượng bán trong khoảng |  |
| 9 | txtNgayBanTu  txtNgayBanDen | Textbox | Tìm kiếm theo ngày bán trong khoảng |  |
| 10 | txtDonGiaTu  txtDonGiaDen | Textbox | Tìm kiếm theo đơn giá trong khoảng |  |
| 11 | gridDanhSachSach | GridControl | Lưới thể hiện thông tin các đầu sách tra cứu thỏa với điều kiện với các thông tin chi tiết liên quan: MaSach, TenSach, TheLoai, TacGia, SoLuongTon, SoLuongBan,  TongThanhTien |  |
| 12 | btnTraCuu | Button | Nút này cho phép tra cứu các sách với các điều kiện | **Xử lý chinh** |
| 13 | btnThoat | Button | Nút này cho phép thoát màn hình tra cứu sách |  |

### Màng hình “Lập phiếu thu tiền”

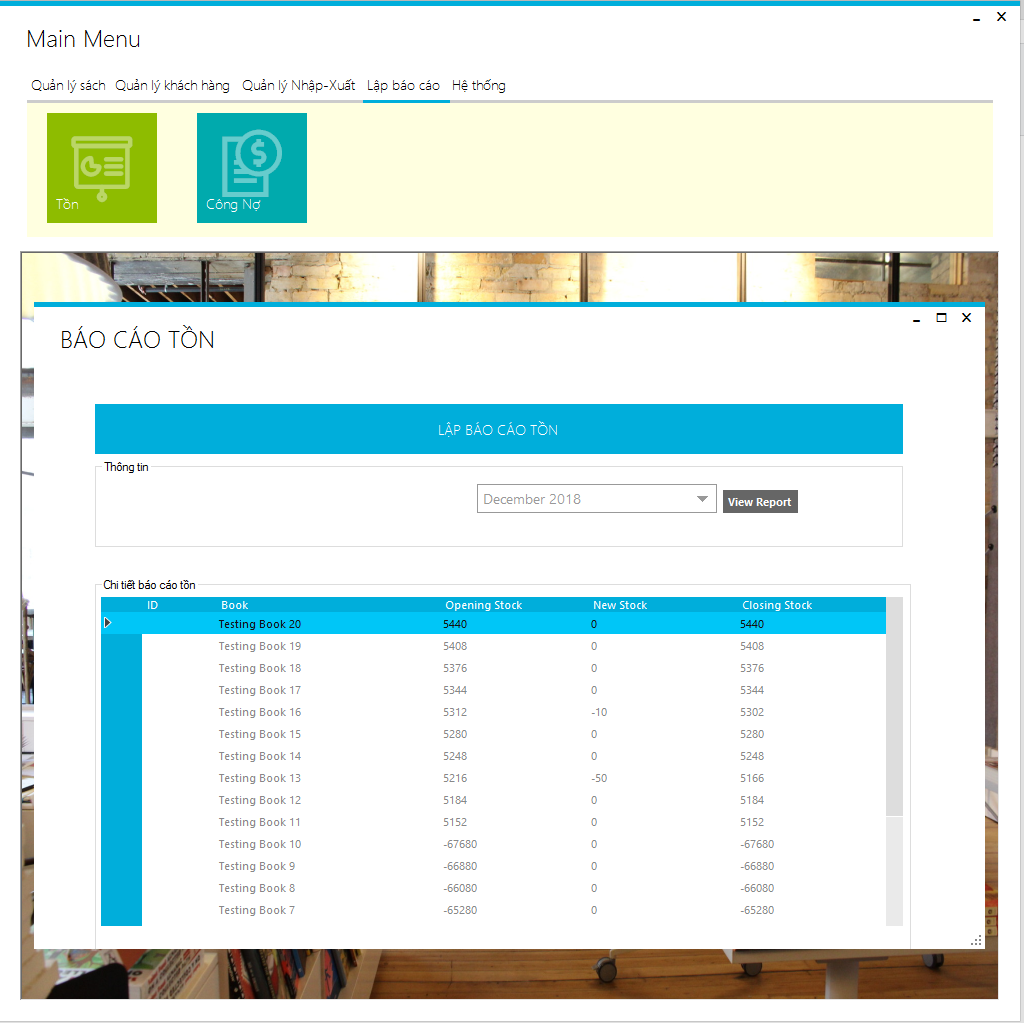
****

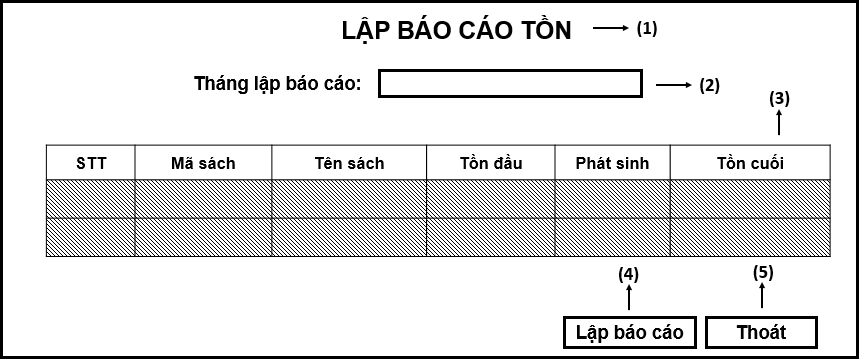


**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblLapPhieuThuTien | Label | Tiêu đề của màn hình lập phiếu thu tiền |  |
| 2 | lxtMaPhieuThu | Textbox | Mã của phiếu thu tiền |  |
| 3 | txtNgayThuTien | Textbox | Ngày thu tiền |  |
| 4 | txtMaKhachHang | Textbox | Mã khách hàng |  |
| 5 | btnTimKhachHang | Button | Hiện thị màn hình Tìm kiếm khách hàng để chọn  khách hàng thu tiền |  |
| 6 | btnHoTenKhachHang | Textbox | Họ tên của khách hàng thu tiền |  |
| 7 | txtDienThoai | Textbox | Điện thoại của khách hàng thu tiền |  |
| 8 | txtDiaChi | Textbox | Địa chỉ của khách hàng thu tiền |  |
| 9 | txtSoTienNo | Textbox | Số tiền khách hàng đang nợ |  |
| 10 | chkApDungQuyDinh | Checkbox | Áp dụng quy định số tiền thu không vượt quá số tiền đang nợ |  |
| 11 | txtSoTienThu | Textbox | Số tiền thu |  |
| 12 | btnTaoMoi | Button | Tạo mới phiếu thu tiền |  |
| 13 | btnHuyPhieuThuTIen | Button | Hủy phiếu thu tiền |  |
| 14 | btnLapPhieuThuTien | Button | Lập phiếu thu tiền | **Xử lý chinh** |
| 15 | btnThoat | Button | Thoát khỏi màn hình lập phiếu thu tiền |  |

### Màng hình “lập báo cáo tồn”

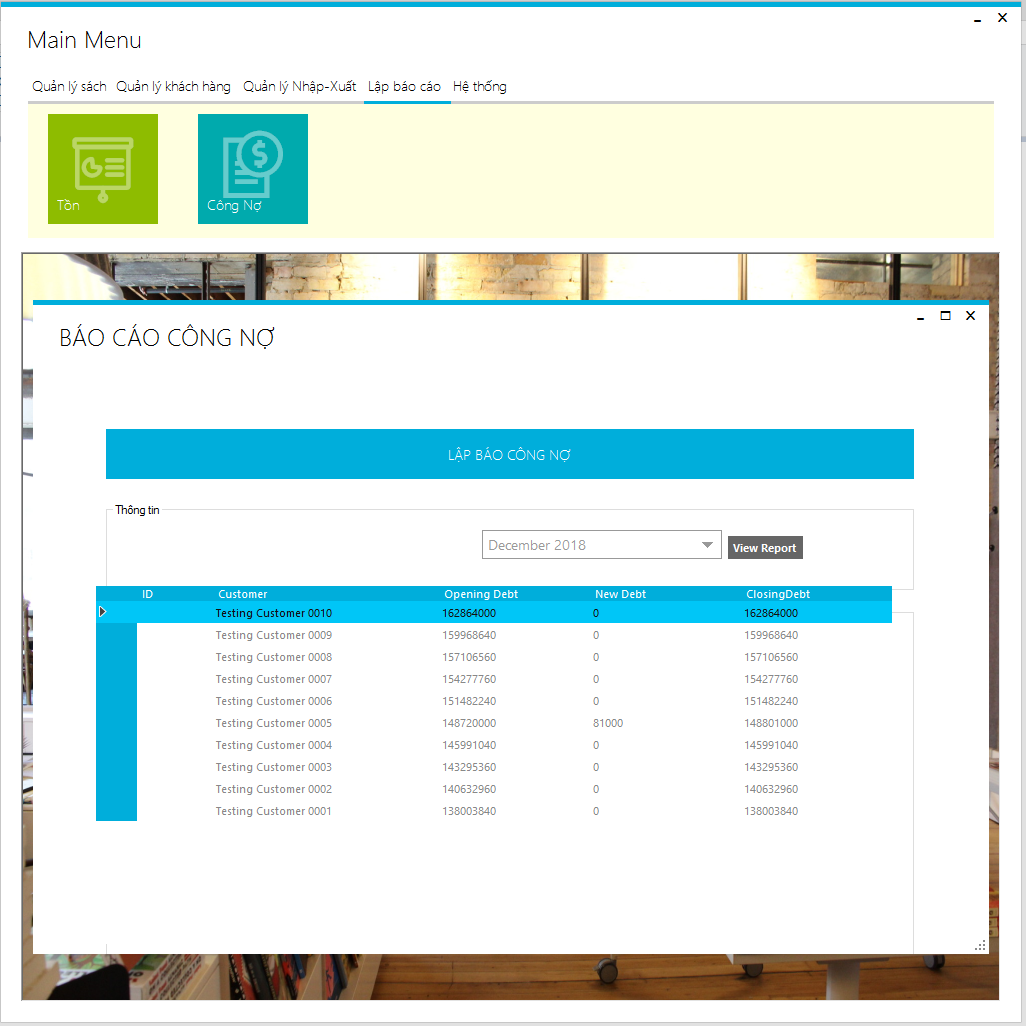




**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblLapBaoCaoTon | Label | Tiêu đề của màn hình lập báo cáo tồn |  |
| 2 | lxtThangBaoCao | Textbox | Tháng của báo cáo tồn |  |
| 3 | gridDanhSachSach | GridControl | Lưới thể hiện thông tin các đầu sách trong tháng báo cáo với các thông tin chi tiết liên quan: Mã sách, Tên sách, Tồn đầu, Phát sinh, Tồn cuối |  |
| 4 | btnLapBaoCao | Button | Lập báo cáo tồn của tháng trong báo cáo | **Xử lý chinh** |
| 5 | btnThoat | Button | Thoát khỏi màn hình lập báo cáo tồn |  |

### Màng hình “Báo cáo công nợ”

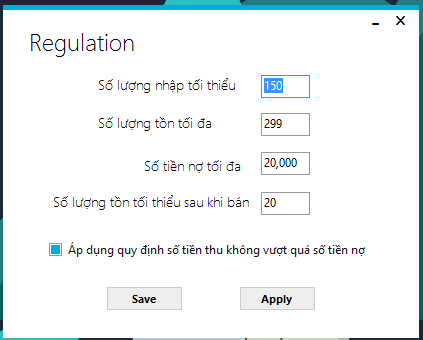
****

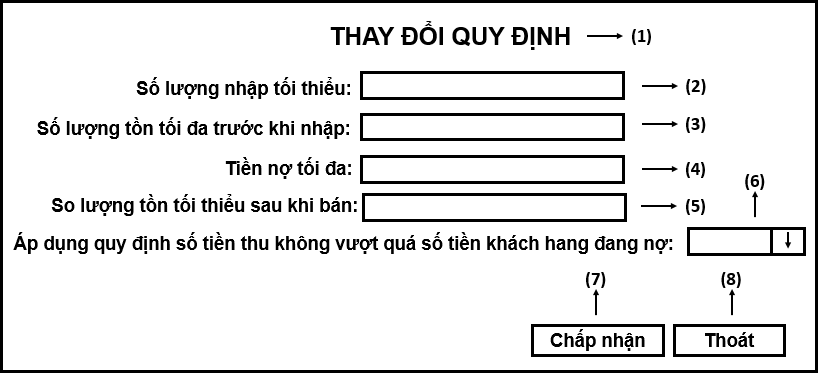


**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblLapBaoCaoCongNo | Label | Tiêu đề của màn hình lập báo cáo công nợ |  |
| 2 | lxtThangBaoCao | Textbox | Tháng báo cáo công nợ |  |
| 3 | gridDanhSachKhachHang | GridControl | Lưới thể hiện thông tin các khách hàng trong tháng báo cáo với các thông tin chi tiết liên quan: Mã khách hàng, Tên khách hàng, Nợ đầu,  Phát sinh, Nợ cuối |  |
| 4 | btnLapBaoCao | Button | Lập báo cáo công nợ của tháng trong báo cáo | **Xử lý chinh** |
| 5 | btnThoat | Button | Thoát khỏi màn hình lập báo cáo tồn |  |

### Màng hình “Thay đổi quy định”

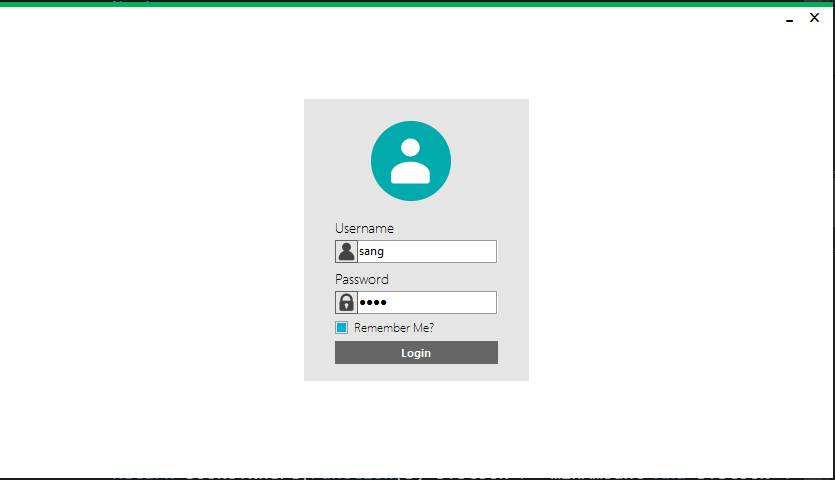
****



**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | lblThayDoiQuyDinh | Label | Tiêu đề của màn  hình thay đổi quy định |  |
| 2 | lxtSoLuongNhapToiThieu | Textbox | Quy định số lượng nhập tối  thiểu |  |
| 3 | txtSoLuongTonToiDaTruocKhiNhap | Textbox | Quy định về số lượng tồn tối đa trước khi nhập |  |
| 4 | txtSoTienNoToiDa | Textbox | Quy định về số tiền nợ tối đa |  |
| 5 | txtSoLuongTonToiThieuSauKhiBan | Textbox | Quy định về số lượng tồn tối thiểu sau khi bán |  |
| 6 | cbApDungQuyDinhTienThuKhong  VuotTienNo | Combobox  True/False | Áp dnjg quy đinh tiền thu không vượt tiền nợ |  |
| 7 | btnChapNhan | Button | Lưu những thay đổi về quy định | **Xử lý**  **chinh** |
| 8 | btnThoat | Button | Thoát khỏi màn hình thay đổi quy  định |  |

### Màng hình đăng nhập

****

**Danh sách các đối tượng thể hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ý nghĩa** | **Ghi chú** |
| 1 | txtTenDangNhap | Textbox | Lấy tên đăng nhập từ người dùng |  |
| 2 | txtMatKhau | Textbox | Lấy mật khẩu từ người dùng |  |
| 3 | btnDangNhap | Button | Kiểm tra thông tin và cho phép đăng nhập hoặc yêu cầu nhập lại |  |
| 4 | btnHuyBo | Button | Thoát khỏi ứng dụng |  |

## CHƯƠNG 5: TỔNG KẾT

**I – Môi trường phát triển và môi trường triển khai**

* Công cụ xây dựng ứng dụng: Visual Studio
* Hệ quản trị cơ sở dữ liệu: SQL Server
* Công cụ vẽ sơ đồ: Microsoft Visio
* UI Framework: Metro Framework

**II – Kết quả đạt được**

* Phần mềm đáp ứng đầy đủ các chức năng cần thiết cho một phần mềm quản lý nhà sách.
* Tin học hóa được hầu hết các nghiệp vụ.
* Giao diện thân thiện, dễ dùng.
* Ứng dụng được xây dựng theo mô hình 3 lớp, dễ phát triển và sửa lỗi.

**III – Hướng phát triển**

* Cải thiện chống SQL injection
* Cải thiện chức năng tìm kiếm nhiều trường dữ liệu hơn
* Tích hợp thêm nhiều chức năng hỗ trợ quản lý nhân viên như chấm công, tính lương.
* Phát triển cho phép thành viên là khách hàng đăng nhập và tìm kiếm sách online, cũng như review trước khi mua sách.
* Tích hợp input là máy đọc mã vạch.

**IV – Phân công công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Họ tên | MSSV | Mức độ hoàn thành | Công việc được giao | Ký tên |
| Nguyễn Huỳnh Sáng | 16521022 | 100% | - Nhóm trưởng, phân công công việc  - Xây dựng một số chức năng:  + Liên quan tới Báo cáo  + Liên quan tới Quản lý xuất nhập  + Thay đổi quy định  + Thiết kế UI  - Viết báo cáo |  |
| Phan Hoàng Nhân | 16520873 | 100% | - Xây dựng một số chức năng:  + Liên quan tới Sách  + Phân quyền  + Đăng nhập  + Khách hang  + Thiết kế UI  - Thiết kế màn hình welcome |  |
| Hồng Gia Lập | 16520654 | 100% | Thiết kế database, các lớp DTO, DAL, BUS  - Xây dựng một số chức năng:  + Liên quan tới Báo cáo |  |

## CHƯƠNG 6: TÀI LIỆU THAM KHẢO

[1]. Slide bài giảng

[2]. https://vi.wikipedia.org/