

**TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI**  
**KHOA CÔNG NGHỆ THÔNG TIN**

----



**Tiểu luận: Công nghệ phần mềm**  
**Xây dựng biểu đồ Use Case**

Nhóm Thực Hành : 9

Tên lớp học phần : 68CS3

Giảng Viên : Lê Thị Hoàng Anh

Thành viên nhóm :  
Đỗ Công Trí  
Nguyễn Đức Mạnh  
Lâm Tuấn Kiệt  
Nguyễn Phúc Tiến  
Trần Phương Nam

## Lời mở đầu

Trong quá trình phát triển phần mềm, việc phân tích và thiết kế hệ thống là một bước quan trọng nhằm đảm bảo rằng phần mềm được xây dựng đúng yêu cầu và dễ bảo trì trong tương lai. Một trong những công cụ mạnh mẽ hỗ trợ quá trình này là UML (Unified Modeling Language) – ngôn ngữ mô hình hóa thống nhất. UML 2.0 cung cấp một bộ ký hiệu và sơ đồ giúp các nhà phân tích, thiết kế và lập trình viên có thể mô tả, trực quan hóa và xây dựng hệ thống một cách hiệu quả.

Trong số các loại biểu đồ mà UML hỗ trợ, biểu đồ UseCase đóng vai trò quan trọng trong việc mô tả các chức năng của hệ thống dưới góc nhìn của người dùng. Biểu đồ UseCase giúp xác định rõ các tác nhân tương tác với hệ thống và những gì hệ thống cần thực hiện để đáp ứng nhu cầu đó.

Bên cạnh lý thuyết về UML, việc sử dụng các công cụ hỗ trợ như Visual Paradigm sẽ giúp việc xây dựng biểu đồ UseCase trở nên trực quan, chính xác và dễ dàng chia sẻ trong nhóm phát triển phần mềm.

Trong bài viết này, chúng em sẽ trình bày tổng quan về UML 2.0, tập trung vào biểu đồ UseCase và công cụ Visual Paradigm. Đồng thời, bài viết sẽ đưa ra ví dụ minh họa cụ thể, cũng như xây dựng biểu đồ UseCase cho chức năng liệt kê danh sách các nguyên liệu cần nhập bổ sung vào kho hàng – một chức năng phổ biến trong các hệ thống quản lý kho.

Trân trọng cảm ơn!

## MỤC LỤC

<b>I. Tìm hiểu UML 2.0, công cụ Visual Paradigm cho biểu đồ UseCase. Cho ví dụ minh họa bất kỳ. ..</b>	<b>5</b>
<b>1. UML 2.0 .....</b>	<b>5</b>
<b>2. Biểu đồ Use Case.....</b>	<b>5</b>
2.1. Biểu đồ UC bao gồm:.....	6
2.2. Các mối quan hệ trong Use Case: .....	6
2.3. Mục đích biểu đồ Use Case .....	7
<b>3. Công cụ Visual Paradigm .....</b>	<b>8</b>
<b>4. Ví dụ về Use Case Diagram ( Ứng dụng ATM ).....</b>	<b>9</b>
4.1. Actors .....	9
4.2. Use Case .....	9
4.3. Xác định các quan hệ .....	10
<b>II. Vẽ biểu đồ UseCase để thực hiện được chức năng liệt kê danh sách các nguyên liệu cần nhập bổ sung vào kho hàng. ....</b>	<b>11</b>
<b>1. Phân tích hệ thống .....</b>	<b>11</b>
<b>2. Xây dựng.....</b>	<b>12</b>

## BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Nhiệm vụ	Người phụ trách
1	Tìm hiểu lý thuyết UML 2.0 và biểu đồ Use Case	Trần Phương Nam
2	Phân tích hệ thống quản lý bán đồ uống sữa hạt, xác định Actor và Use Case	Nguyễn Đức Mạnh
3	Nghiên cứu công cụ Draw.io	Nguyễn Phúc Tiến
4	Vẽ biểu đồ Use Case cho chức năng liệt kê nguyên liệu cần nhập kho	Đỗ Công Trí
5	Viết báo cáo và hoàn thiện tài liệu	Lâm Tuấn Kiệt
6	Kiểm tra, chỉnh sửa	Cả nhóm

# I. Tìm hiểu UML 2.0, công cụ Visual Paradigm cho biểu đồ UseCase. Cho ví dụ minh họa bất kỳ.

## 1. UML 2.0

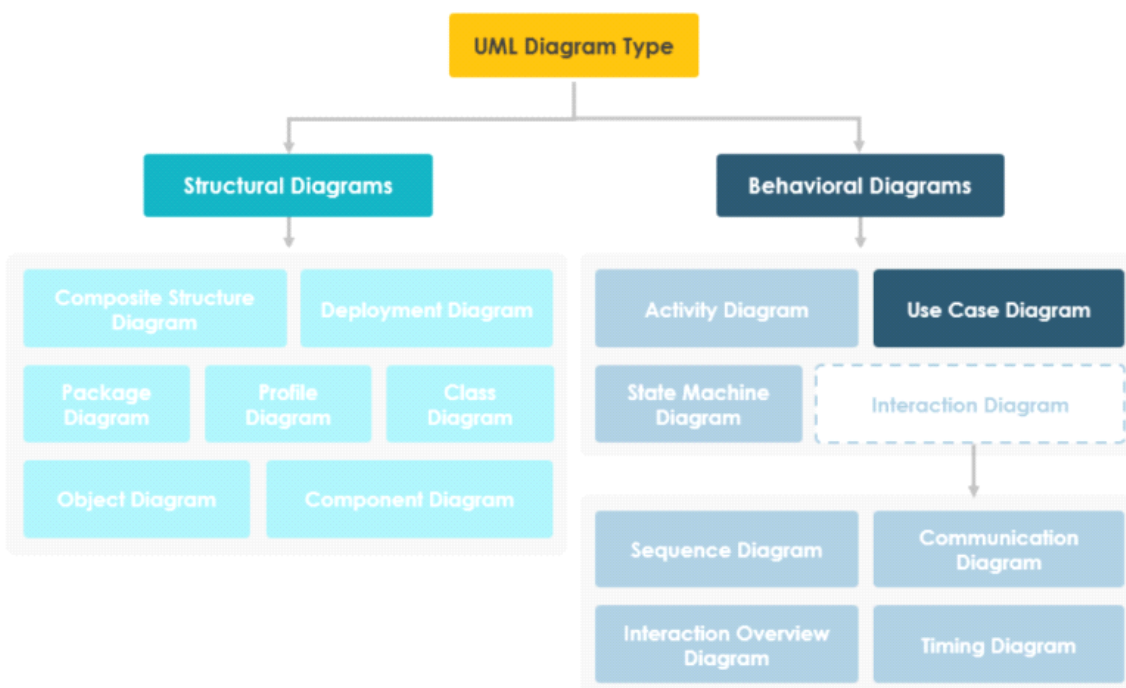
UML (Unified Modeling Language) là ngôn ngữ mô hình hóa thống nhất, được sử dụng để phân tích, thiết kế và mô tả hệ thống phần mềm theo hướng trực quan. UML giúp các nhà phát triển, nhà phân tích và khách hàng có cùng một cách nhìn chung về hệ thống. Phiên bản UML 2.0 ra đời nhằm chuẩn hóa và mở rộng UML, bao gồm 13 loại biểu đồ được chia thành hai nhóm chính:

- Biểu đồ cấu trúc (Structural diagrams): Class Diagram, Object Diagram, Component Diagram, Deployment Diagram, Package Diagram, Composite Structure Diagram.
- Biểu đồ hành vi (Behavioral diagrams): Use Case Diagram, Sequence Diagram, Activity Diagram, State Machine Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.

UML 2.0 đặc biệt được ứng dụng trong phát triển phần mềm hướng đối tượng, giúp giảm thiểu sai sót trong quá trình thu thập yêu cầu, thiết kế và hiện thực hóa phần mềm

## 2. Biểu đồ Use Case

Biểu đồ Use Case là một trong những biểu đồ quan trọng của UML, được sử dụng để mô tả các chức năng của hệ thống dưới góc nhìn của người dùng (actor).



## 2.1. Biểu đồ UC bao gồm:

- Thành phần của biểu đồ Use Case (UC Diagram):
- Hệ thống: Thể hiện bằng một hình chữ nhật bao quanh các chức năng (Use Case) bên trong. Đây chính là phạm vi của hệ thống cần xây dựng.

Biểu đồ UC giống như “bản đồ tổng quan” của hệ thống, thể hiện:

Điều gì đang được mô tả? (Hệ thống)

Khung chữ nhật trong sơ đồ chính là ranh giới hệ thống.

Nó giúp người xem biết phạm vi của hệ thống: hệ thống làm gì và không làm gì.

Ví dụ: Trong hệ thống quản lý phòng họp, phạm vi hệ thống chỉ gồm đặt phòng, quản lý thiết bị, phê duyệt đăng ký..., còn những việc như bảo trì phòng họp có thể nằm ngoài.

- Tác nhân (Actor): Là người hoặc hệ thống bên ngoài có tương tác với hệ thống (ví dụ: khách hàng, nhân viên, quản trị viên).

Actor được vẽ bên ngoài hệ thống, cho thấy người dùng nào sẽ tương tác với hệ thống.

Điều này giúp nhóm phát triển xác định đúng đối tượng cần phục vụ (khách hàng, nhân viên, quản trị viên,...).

Actor muốn đạt được :

Mỗi Use Case là một mục tiêu hay nhu cầu của actor.

Nó thể hiện kỳ vọng của người dùng khi làm việc với hệ thống.

Ví dụ: Actor “Nhân viên” muốn “Đăng ký phòng họp”, actor “Quản trị viên” muốn “Phê duyệt đăng ký”.

- Use Case (UC / Chức năng): Là những chức năng hay dịch vụ mà hệ thống cung cấp để đáp ứng nhu cầu của actor (ví dụ: đăng nhập, đăng ký tài khoản, đặt phòng).

## 2.2. Các mối quan hệ trong Use Case:

- Association: quan hệ giữa Actor và Use Case.

- Include: một Use Case luôn bao hàm chức năng của Use Case khác.
- Extend: một Use Case có thể mở rộng thêm chức năng trong điều kiện.
- Generalization: quan hệ kế thừa giữa Actor hoặc Use Case

## 2.3. Mục đích biểu đồ Use Case

### 1. Xác định bối cảnh (System Context)

Biểu đồ UC cho thấy ranh giới hệ thống: cái gì nằm trong hệ thống và cái gì nằm ngoài.

Giúp phân biệt rõ chức năng hệ thống cần làm và chức năng bên ngoài (actor).

### 2. Nắm bắt yêu cầu chức năng của hệ thống (Functional Requirements)

UC được viết từ góc nhìn người dùng, nên nó tập trung vào hệ thống phải cung cấp những gì, không đi sâu vào cách triển khai giúp Người dùng (không rành kỹ thuật) vẫn hiểu được nhà phát triển (dev) biết rõ phải làm gì.

Ví dụ: Actor “Người quản trị” có UC “Quản lý thiết bị”, “Phân quyền người dùng”.

### 3. Giao tiếp giữa các bên liên quan (Stakeholder Communication)

Biểu đồ UC là “ngôn ngữ chung” giữa khách hàng, người dùng, nhà phân tích nghiệp vụ, tester và lập trình viên nó giúp giảm hiểu lầm vì mô tả trực quan, ai cũng dễ hình dung.

Ví dụ: Thay vì mô tả dài dòng, chỉ cần chỉ vào sơ đồ: “Người dùng A có quyền Đặt phòng và Hủy phòng, còn Người quản trị có quyền Duyệt phòng”.

### 4. Xác thực kiến trúc ở mức cao (High-level Architecture Validation)

UC thể hiện tổng quan tất cả các chức năng chính của hệ thống, do đó có thể kiểm tra sớm:

- Hệ thống có đủ chức năng không?
- Có chức năng nào thừa hoặc sai yêu cầu không?

Giúp tiết kiệm chi phí vì phát hiện sai sót ở giai đoạn sớm thay vì khi đã code. Làm cơ sở cho phân tích và thiết kế chi tiết

### 5. Xác định lớp (Class) và luồng dữ liệu.

Hỗ trợ kiểm thử (Test Case Generation)

Mỗi UC thường được chuyển thành kịch bản kiểm thử (test case).

Ví dụ: UC “Đặt phòng” → Test case 1: Đặt phòng thành công, Test case 2: Đặt phòng trùng lịch, Test case 3: Người không có quyền cố gắng đặt phòng.

Điều này đảm bảo rằng khi test, hệ thống thực sự đáp ứng được đúng yêu cầu người dùng.

#### 6. Xác định phạm vi hệ thống (Project Scope)

UC giúp xác định hệ thống cần làm tới đâu, tránh việc “scope creep” (mở rộng chức năng ngoài yêu cầu).

### 3. Công cụ Visual Paradigm

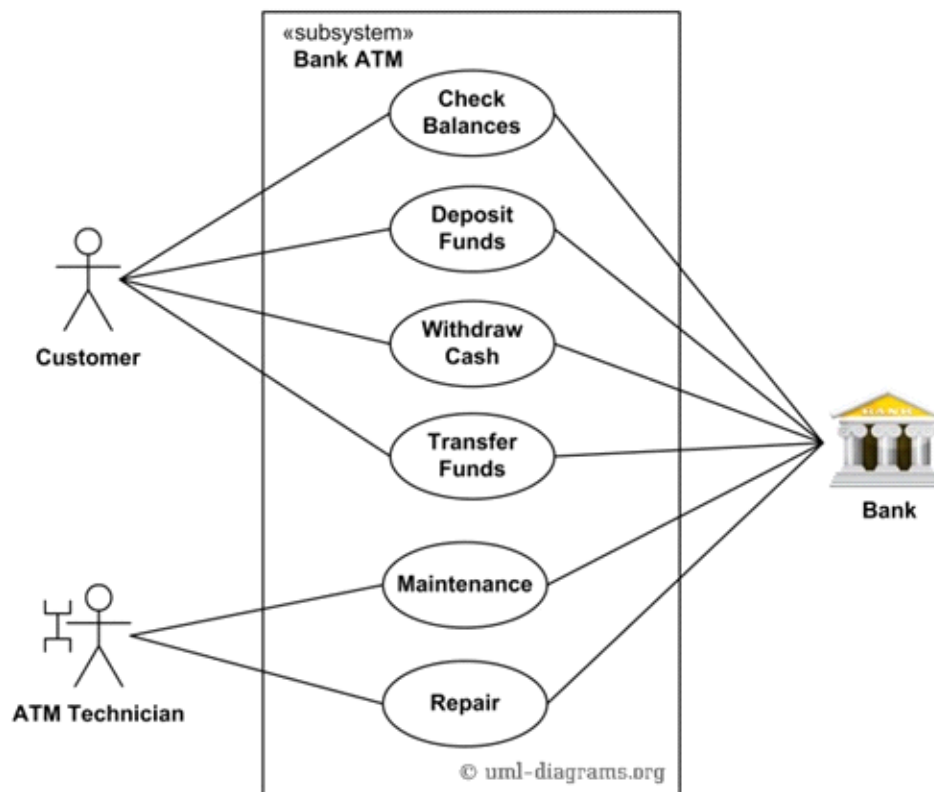
Visual Paradigm là một công cụ mô hình hóa mạnh mẽ, hỗ trợ hầu hết các loại biểu đồ UML, trong đó có Use Case Diagram. Đây là phần mềm phổ biến trong giảng dạy cũng như trong thực tế phát triển phần mềm.

Các tính năng chính của Visual Paradigm:

- Cho phép vẽ biểu đồ UML bằng thao tác kéo – thả trực quan.
- Quản lý nhiều loại biểu đồ trong cùng một dự án.
- Hỗ trợ cộng tác nhóm và quản lý yêu cầu phần mềm.
- Xuất biểu đồ dưới dạng hình ảnh (PNG, JPG) hoặc tài liệu (Word, PDF).



#### 4. Ví dụ về Use Case Diagram ( Ứng dụng ATM )



##### 4.1. Actors

ví dụ về ATM ở trên chúng ta thấy:

- Ai sử dụng hệ thống? -> Customer, ATM Technician
- Hệ thống nào tương tác với hệ thống này? -> Bank

Như vậy có 03 Actor: Customer, ATM Technician và Bank

Các Actor trong hệ thống ATM:

- Customer (khách hàng)
- ATM Technician (kỹ thuật viên ATM)
- Bank (ngân hàng) – là hệ thống hoặc tổ chức bên ngoài tương tác với hệ thống ATM.

##### 4.2. Use Case

- Customer sử dụng các chức năng: Check Balance, Deposit, Withdraw và Transfer

- ATM technician sử dụng: Maintenance và Repair
- Bank tương tác với tất cả các chức năng trên.

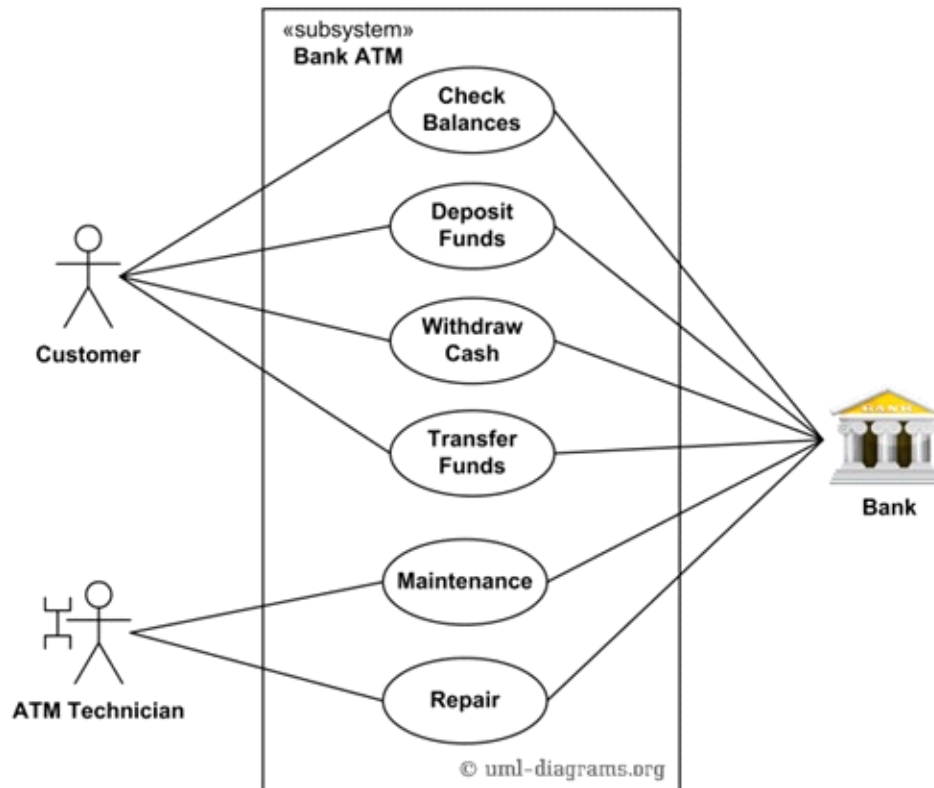
Tóm lại, chúng ta phải xây dựng hệ thống có các chức năng: Check Balance, Deposit, Withdraw, Transfer, Maintenance và Repair để đáp ứng được cho người sử dụng và các hệ thống tương tác.

Các Use Case (chức năng) trong hệ thống:

- Check Balance (Kiểm tra số dư)
- Deposit (Nạp tiền)
- Withdraw (Rút tiền)
- Transfer (Chuyển khoản)
- Maintenance (Bảo trì)
- Repair (Sửa chữa)

#### 4.3. Xác định các quan hệ

Phân tích và xác định các quan hệ giữa các Actor và Use Case, giữa các Actor với nhau, giữa các Use Case với nhau sau đó nối chúng lại chúng ta sẽ được bản vẽ Use Case.



Bank tương tác với tất cả các chức năng do các Actor khác thực hiện, nghĩa là:

- Bank tham gia vào việc xử lý các giao dịch như Check Balance, Deposit, Withdraw, Transfer:  
Đây là các chức năng tài chính mà ngân hàng quản lý dữ liệu và thực hiện giao dịch thật.
- Bank cũng hỗ trợ hoặc giám sát quá trình Maintenance và Repair thông qua hệ thống kỹ thuật.

Các quan hệ cụ thể của Bank trong Use Case Diagram:

- Association (liên kết) giữa Bank và tất cả các Use Case trên, thể hiện Bank là hệ thống tương tác, cung cấp dữ liệu, xử lý giao dịch hoặc hỗ trợ kỹ thuật.
- Quan hệ giữa Bank và các Use Case khác là association, bởi Bank là đối tượng ngoài hệ thống (Actor), không phải Use Case.

Qua việc phân tích quan hệ của các Actor trong hệ thống ATM, đặc biệt là Bank, chúng ta nhận thấy Bank đóng vai trò quan trọng trong việc tương tác và hỗ trợ các chức năng chính của hệ thống như kiểm tra số dư, nạp tiền, rút tiền, chuyển khoản cũng như bảo trì và sửa chữa. Việc xác định rõ các quan hệ này giúp xây dựng và thiết kế hệ thống một cách chính xác, đáp ứng đúng yêu cầu của người dùng và các bên liên quan.

## **II. Vẽ biểu đồ UseCase để thực hiện được chức năng liệt kê danh sách các nguyên liệu cần nhập bổ sung vào kho hàng.**

**Đề tài: Xây dựng chương trình quản lý bán đồ uống sữa hạt (sữa có nguồn gốc từ các loại hạt thực vật)**

### **1. Phân tích hệ thống**

Trong chương trình quản lý bán đồ uống sữa hạt, chức năng quan trọng là theo dõi và bổ sung nguyên liệu vào kho. Để thực hiện chức năng này, ta xác định:

Các tác nhân (Actor):

- Nhân viên kho: người kiểm tra, theo dõi tình trạng nguyên liệu và lập danh sách cần nhập.
- Quản lý: người duyệt và quyết định nhập hàng.

Các chức năng (Use Case):

- Kiểm tra tồn kho: kiểm tra số lượng nguyên liệu còn lại trong kho.
- Liệt kê danh sách nguyên liệu cần nhập: lập danh sách các nguyên liệu sắp hết hoặc đã hết.
- Lập phiếu đề nghị nhập nguyên liệu: tạo phiếu đề nghị bổ sung nguyên liệu gửi quản lý.
- Phê duyệt phiếu nhập: quản lý xem xét và phê duyệt đề nghị nhập nguyên liệu.

## **2. Xây dựng**

Xây dựng Use Case cho hệ thống quản lý bán đồ uống sữa hạt

### **Bước 1. Thu thập kiến thức liên quan đến hệ thống**

Hệ thống quản lý bán đồ uống sữa hạt không chỉ phục vụ bán hàng mà còn phải đảm bảo kho nguyên liệu luôn đủ để sản xuất và kinh doanh.

Đặc thù nguyên liệu:

- Các loại hạt (hạnh nhân, óc chó, điều, đậu nành, yến mạch, mắc ca, hạt sen, )
- Nguyên liệu phụ (đường, muối, hương liệu, bao bì, chai lọ).
- Một số nguyên liệu có hạn sử dụng ngắn → phải theo dõi chặt chẽ.

Hoạt động thực tế:

- Nhân viên kho kiểm kê số lượng nguyên liệu định kỳ hoặc khi có yêu cầu.
- Khi phát hiện nguyên liệu sắp hết, nhân viên kho phải lập danh sách nguyên liệu cần nhập bổ sung.
- Danh sách này được gửi lên quản lý để xem xét và phê duyệt.
- Nếu được duyệt, hệ thống sẽ sinh ra phiếu nhập nguyên liệu gửi cho nhà cung cấp.

Mục tiêu của chức năng Use Case:

- Đảm bảo kho luôn đủ nguyên liệu để duy trì hoạt động bán hàng, hạn chế rủi ro thiếu nguyên liệu làm gián đoạn kinh doanh.

### **Bước 2. Xác định Actor**

Hệ thống :

Nhân viên kho

- Kiểm tra tồn kho
- Lập danh sách nguyên liệu cần nhập
- Lập phiếu đề nghị nhập

Quản lý

- Xem danh sách do nhân viên kho lập.
- Phê duyệt hay từ chối phiếu nhập nguyên liệu.
- Hệ thống quản lý bán hàng (Inventory System – optional)

### **Bước 3. Xác định Use Case**

Chức năng :

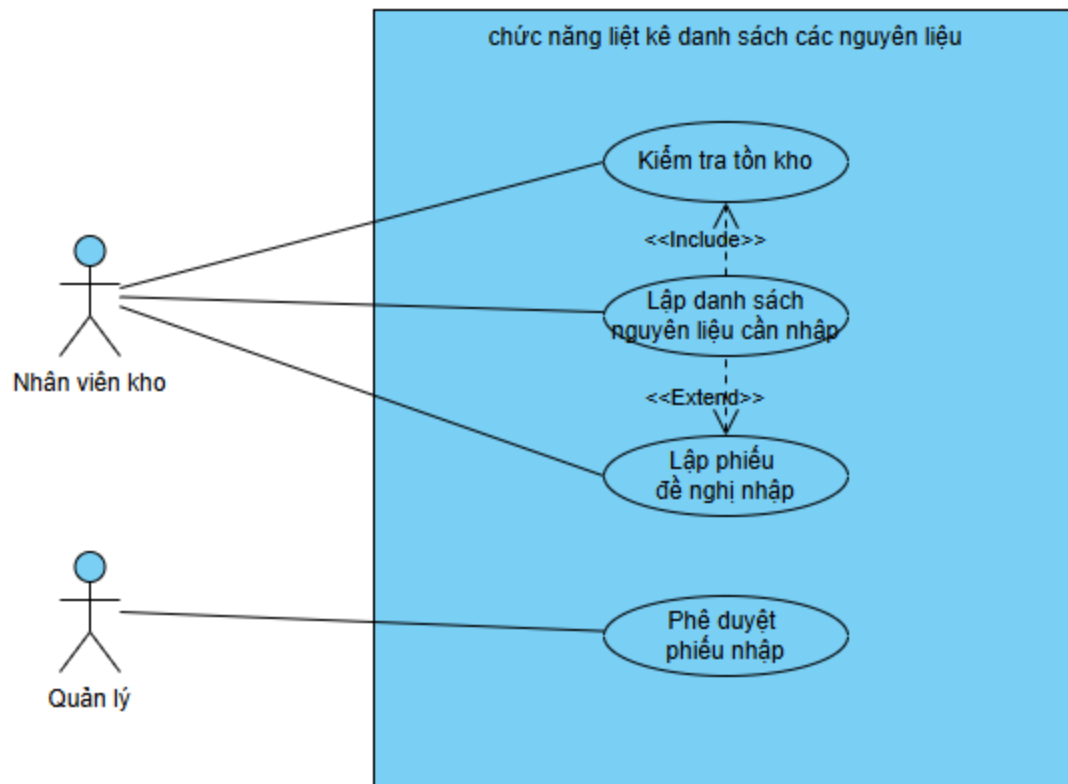
Với Nhân viên kho

- Kiểm tra tồn kho (Inventory Check)  
→ Xem số lượng nguyên liệu còn lại trong kho.
- Liệt kê danh sách nguyên liệu cần nhập (List Materials to Restock)  
→ Từ dữ liệu tồn kho, xác định loại nguyên liệu sắp hết.
- Lập phiếu đề nghị nhập nguyên liệu (Create Purchase Request)  
→ Gửi danh sách nguyên liệu cần nhập lên quản lý.

Với Quản lý

- Phê duyệt phiếu nhập (Approve Purchase Request)  
→ Quyết định nhập hay không nhập thêm nguyên liệu.

### **Bước 4. Vẽ bản vẽ Use Case (quan hệ giữa các chức năng)**



Các quan hệ chính:

- Association (kết nối Actor – Use Case):
- Nhân viên kho ↔ Kiểm tra tồn kho
- Nhân viên kho ↔ Liệt kê danh sách nguyên liệu cần nhập
- Nhân viên kho ↔ Lập phiếu đề nghị nhập
- Quản lý ↔ Phê duyệt phiếu nhập

Include (bắt buộc):

- Liệt kê danh sách nguyên liệu cần nhập → include → Kiểm tra tồn kho

(Muốn liệt kê thì phải kiểm tra kho trước).

Extend (mở rộng):

- Liệt kê danh sách nguyên liệu cần nhập → extend → Lập phiếu đề nghị nhập

(Sau khi liệt kê, có thể lập phiếu để gửi quản lý, nhưng không bắt buộc ngay).

Hình dung sơ đồ:

- Actor Nhân viên kho kết nối với 3 Use Case.
- Actor Quản lý kết nối với “Phê duyệt phiếu nhập”.
- Giữa các Use Case có quan hệ include và extend như đã mô tả.

### **Bước 5. Giải thích quan hệ**

- Association: Cho biết actor nào được phép thực hiện chức năng nào.
- Include: Cho biết chức năng nào phải thực hiện trước để hỗ trợ chức năng khác. Ví dụ: muốn liệt kê thì buộc phải kiểm tra tồn kho.
- Extend: Cho biết chức năng nào có thể mở rộng thêm sau khi thực hiện một chức năng khác. Ví dụ: sau khi liệt kê danh sách thì có thể mở rộng bằng cách lập phiếu nhập.

### **Bước 6. Kết luận**

Biểu đồ Use Case cho chức năng “Liệt kê danh sách nguyên liệu cần nhập bổ sung vào kho hàng” đã thể hiện:

- Nhân viên kho đóng vai trò trung tâm: kiểm tra kho → liệt kê → có thể lập phiếu nhập.
- Quản lý là người có quyền quyết định cuối cùng bằng cách phê duyệt phiếu nhập.
- Các quan hệ include và extend đảm bảo Use Case được mô tả logic, theo đúng thực tế quy trình nghiệp vụ.

## **Lời cảm ơn**

Trước hết, nhóm xin gửi lời cảm ơn chân thành đến cô Lê Thị Hoàng Anh đã tận tình giảng dạy, định hướng và tạo điều kiện thuận lợi để chúng em có thể tiếp cận, nghiên cứu và hoàn thành bài tiểu luận này.

Nhóm cũng xin cảm ơn các thành viên trong nhóm đã luôn nỗ lực, phối hợp, phân công công việc hợp lý và hỗ trợ lẫn nhau trong suốt quá trình tìm hiểu và thực hiện đề tài.

Mặc dù đã rất cố gắng, bài tiểu luận khó tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự góp ý từ thầy/cô và các bạn để nội dung được hoàn thiện hơn.

Xin chân thành cảm ơn!



## Tài liệu tham khảo

- Object Management Group (OMG). Unified Modeling Language (UML) Specification Version 2.5.1. 2017. Truy cập tại: <https://www.omg.org/spec/UML/2.5.1>
- Visual Paradigm. Visual Paradigm User Guide: Use Case Diagram. Truy cập tại: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
- Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. Prentice Hall, 2004.
- Fowler, Martin. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition. Addison-Wesley, 2003.
- Ambler, Scott W. The Elements of UML 2.0 Style. Cambridge University Press, 2005.