

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

# Project Proposal



Bộ môn Công nghệ phần mềm  
Khoa Công nghệ thông tin  
Đại học Khoa học tự nhiên TP HCM

# MỤC LỤC

Các nội dung chính.....	1
1 Thông tin các thành viên.....	2
2 Phát biểu bài toán sơ lược .....	3
3 Giải pháp đề xuất .....	5
4 Kế hoạch phát triển.....	10
5 Kế hoạch nhân sự & chi phí .....	21

# Project Proposal

## Các nội dung chính

Mục tiêu tài liệu tập trung vào các chủ đề:

- ✓ Tạo ra tài liệu Project Proposal.
- ✓ Hoàn chỉnh tài liệu Project Proposal với các nội dung:
  - Phát biểu bài toán sơ lược
  - Giải pháp đề xuất
  - Kế hoạch phát triển
  - Kế hoạch nhân sự & chi phí
- ✓ Đọc hiểu tài liệu Project Proposal.

## 1

## Thông tin các thành viên

MSSV	Họ Tên	Email	% Đóng góp	Chữ ký
22120289	Hồ Ngọc Trung Quân	hntquan0805@gmail.com	100%	
22120304	Nguyễn Thị Kim Quý	quycbq@gmail.com	100%	
22120338	Đỗ Hạnh Thảo	dohanhthao2004@gmail.com	100%	
22120341	Nguyễn Minh Thiện	nguyenminhthien2708@gmail.com	100%	
22120351	Vũ Thị Kim Thoa	ktvu.1404@gmail.com	100%	

# 2

## Phát biểu bài toán sơ lược

Công ty TNHH Phân phối Hàng tiêu dùng An Phú đang gặp phải một vấn đề đau đầu. Là một công ty chuyên cung cấp hàng tiêu dùng cho các đại lý bán lẻ trên khắp cả nước, số lượng đại lý ngày càng tăng khiến việc quản lý trở nên phức tạp.

Trước đây, việc theo dõi thông tin đại lý, danh sách các đại lý tiếp quản, lập phiếu xuất hàng, thu tiền, và báo cáo doanh số, nợ nần đều được thực hiện thủ công. Nhưng với số lượng đại lý lên đến hàng trăm, những cách quản lý truyền thống này đã không còn hiệu quả. Những lỗi sai về số liệu thường xuyên xảy ra, dẫn đến việc các đại lý bị nhầm lẫn trong đơn hàng, nợ không được theo dõi đầy đủ. Thậm chí, có lúc nhân viên của công ty còn bỏ sót phiếu thu tiền. Điều này ảnh hưởng lớn đến dòng tiền và hiệu suất kinh doanh của công ty.

Nhận thấy được vấn đề trên, nhóm 6 đã đề xuất một ứng dụng cho công ty. Ứng dụng này sẽ giúp công ty không chỉ theo dõi thông tin từng đại lý một cách dễ dàng mà còn tự động lập phiếu xuất hàng và thu tiền khi phân phối sản phẩm. Ngoài ra, ứng dụng cũng hỗ trợ báo cáo doanh số và quản lý công nợ của các đại lý, giúp lãnh đạo nắm rõ tình hình tài chính để đưa ra những quyết định kinh doanh chính xác hơn.

➤ Cụ thể:

- Ứng dụng được khởi động đơn giản bằng cách khởi chạy file thực thi của chương trình mà không cần phải cài đặt.
- Khi bắt đầu phần mềm sẽ là một trang chủ với các nút chức năng chính đã nêu ở trên kèm với một số hình ảnh liên quan.
- Người dùng có thể tự do chọn chức năng theo ý muốn bằng cách nhấp chuột, mỗi chức năng sẽ dẫn người dùng tới giao diện tiếp theo cụ thể hơn (như lập phiếu thì sẽ hiện ra là phiếu xuất hàng hay phiếu thu tiền).
- Người dùng có thể chuyển qua lại giữa các chức năng bằng thanh menu.
- Một số tiện ích khác, người dùng có khả năng lưu lại những biểu mẫu mình muốn dưới dạng file hoặc in biểu mẫu.

- Đối với front-end: Sử dụng HTML, CSS và JavaScript để thiết kế giao diện người dùng thân thiện, tương thích trên nhiều thiết bị. Framework hỗ trợ được dùng là Bootstrap.
- Đối với back-end: Sử dụng JavaScript làm nền tảng chính để xây dựng và xử lý các dịch vụ phía máy chủ. Đảm bảo khả năng mở rộng, hiệu năng cao và tích hợp dễ dàng với cơ sở dữ liệu, API và các dịch vụ khác.
- Phần mềm được viết để chạy trên Windows - chỉ hỗ trợ từ Windows 10 trở lên để phần mềm hoạt động ổn định hơn.

# 3 Giải pháp đề xuất

## 3.1 Phần mềm

### 3.1.1. Danh sách các chức năng phần mềm

Nhu cầu	Yêu cầu	Chi tiết
Là quản lý, tôi muốn quản lý việc thêm hoặc xóa tài khoản cho các nhân viên.	Quản lý tài khoản	Nhân viên chưa có tài khoản sẽ thực hiện việc đăng ký tài khoản, sau khi cung cấp đầy đủ các thông tin cần thiết và chọn đăng ký thì hồ sơ của nhân viên sẽ được lưu lại, chờ quản lý duyệt qua, chọn đồng ý lập tài khoản cho nhân viên thì tài khoản sẽ được thiết lập. Chỉ quản lý mới có quyền xóa tài khoản đã đăng ký của nhân viên.
Là người dùng đã đăng ký, tôi muốn đăng nhập vào hệ thống để thao tác tương ứng với quyền hạn của mình (nhân viên hoặc quản lý).	Đăng nhập	Người dùng phải cung cấp tên đăng nhập và mật khẩu tương ứng với tài khoản đăng ký. Nếu người dùng nhập thông tin chính xác thì sẽ được đưa đến trang chủ. Nếu nhập sai thì sẽ báo lỗi.
Là người dùng đã đăng ký, tôi muốn tài khoản của tôi phải có độ bảo mật cao.	Bảo vệ mật khẩu	Mật khẩu người dùng được lưu trữ mã hóa, có độ dài từ 8 ký tự trở lên, phải có ít nhất một ký tự in hoa và 1 ký tự đặc biệt (@, *, #, %,...)

Nhu cầu	Yêu cầu	Chi tiết
Là người dùng, tôi muốn mở rộng quy mô doanh nghiệp bằng cách tiếp nhận quản lý thêm một số đại lý.	Tiếp nhận đại lý	Nhân viên hoặc quản lý phải cung cấp các thông tin: tên đại lý, điện thoại, quận, email, loại đại lý, địa chỉ (do chủ đại lý cung cấp) và ngày tiếp nhận. Hệ thống sẽ ghi nhận và lập hồ sơ đại lý.
Là người dùng đã đăng ký, tôi muốn cung cấp hàng hóa cho các đại lý trong doanh nghiệp và lưu trữ lại trên hệ thống.	Lập phiếu xuất hàng	Nhân viên hoặc quản lý doanh nghiệp phải cung cấp các thông tin gồm: đại lý, ngày lập phiếu và danh sách các mặt hàng (mặt hàng, đơn vị tính, số lượng, đơn giá, thành tiền). Hệ thống sẽ ghi nhận các thông tin trên và lập một phiếu xuất hàng.
Là người dùng đã đăng ký, tôi muốn tra cứu thông tin các đại lý trong doanh nghiệp.	Tra cứu đại lý	Để tra cứu, người dùng phải cung cấp một hoặc một vài thông tin như đại lý, loại, quận, tiền nợ. Hệ thống sẽ liệt kê danh sách các đại lý phù hợp với thông tin được cung cấp ban đầu.
Là người dùng đã đăng ký, tôi muốn thực hiện việc thu tiền mà các đại lý đang nợ.	Lập hóa đơn thu tiền	Người dùng phải cung cấp các thông tin: tên đại lý, điện thoại, địa chỉ, email, ngày thu tiền, số tiền thu. Hệ thống sẽ kiểm tra xem số tiền thu vào có vượt quá số tiền nợ không, nếu hợp lệ (số tiền thu vào không vượt quá tiền nợ) thì sẽ ghi nhận các thông tin đó vào phiếu thu tiền.
Là người dùng đã đăng ký, tôi muốn xem báo cáo doanh thu hàng tháng.	Lập báo cáo tháng	Người dùng phải cung cấp tháng, năm cụ thể. Hệ thống sẽ thống kê số phiếu xuất, tiền nợ đầu và nợ cuối của các đại lý để lập báo cáo doanh số và báo cáo công nợ cho người dùng.
Là quản lý, tôi muốn thay đổi một số quy định về số lượng loại đại lý, mặt hàng,...	Thay đổi quy định	Để thay đổi quy định, quản lý phải sửa lại thông tin phần quy định đó trên hệ thống. Hệ thống sẽ cập nhật những thay đổi đó.

### 3.1.2. Kiến trúc tổng thể phần mềm

Nhóm sẽ xây dựng phần mềm trên hệ điều hành Windows với HTML, CSS và JavaScript để tương thích với framework hỗ trợ là Bootstrap.

Nhóm chọn sử dụng mô hình kiến trúc phân tầng kết hợp với mô hình MVC.

#### Lý do chọn kiến trúc phân tầng (Layered Architecture):

- **Chia hệ thống thành các lớp rõ ràng:** Với các chức năng đa dạng như tiếp nhận đại lý, lập phiếu xuất, lập báo cáo và tra cứu dữ liệu, hệ thống cần được chia thành các lớp như:
  - **Lớp giao diện (Presentation Layer):** Nơi người dùng (quản lý, nhân viên) tương tác với hệ thống thông qua màn hình nhập liệu, báo cáo và tra cứu.
  - **Lớp nghiệp vụ (Business Logic Layer):** Xử lý các quy tắc nghiệp vụ như lập phiếu xuất hàng, thu tiền, thay đổi quy định và các tính toán liên quan đến báo cáo, nợ, quản lý hàng hóa.
  - **Lớp dữ liệu (Data Access Layer):** Tương tác với cơ sở dữ liệu để lưu trữ và truy xuất thông tin như đại lý, phiếu xuất hàng, báo cáo doanh thu và thông tin nợ.
- **Dễ bảo trì và mở rộng:** Khi hệ thống có sự phân chia rõ ràng các chức năng, việc nâng cấp, bảo trì, sửa lỗi hay thêm mới tính năng đều dễ dàng hơn vì mỗi lớp có nhiệm vụ cụ thể, giảm phụ thuộc lẫn nhau.
- **Phù hợp với quy mô doanh nghiệp:** Hệ thống có nhiều nghiệp vụ phức tạp cần được quản lý tốt, và kiến trúc phân tầng sẽ giúp hệ thống dễ dàng mở rộng khi quy mô doanh nghiệp phát triển.

### Lý do chọn Model-View-Controller (MVC):

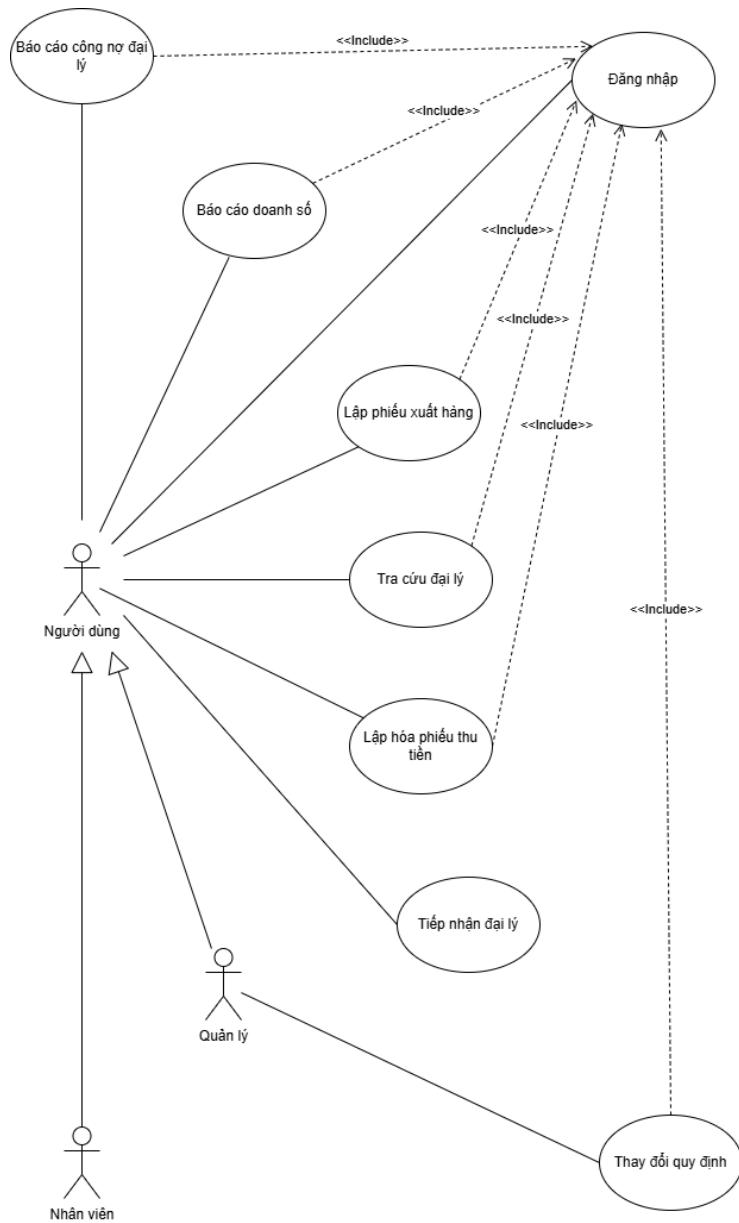
- **Phân chia vai trò rõ ràng giữa giao diện, nghiệp vụ và dữ liệu:** Mô hình MVC giúp phân tách rõ ràng các thành phần chính:
  - **Model component:** Chứa dữ liệu của hệ thống, bao gồm đại lý, phiếu xuất hàng, báo cáo doanh thu, và các thông tin liên quan.
  - **View component:** Là lớp giao diện nơi quản lý và nhân viên tương tác với hệ thống.
  - **Controller component:** Xử lý các yêu cầu từ người dùng và điều hướng chúng đến Model hoặc View tương ứng.
- **Tính linh hoạt và mở rộng dễ dàng:** MVC giúp việc thêm mới các chức năng (ví dụ như thêm tính năng báo cáo mới, thay đổi quy định...) dễ dàng hơn mà không ảnh hưởng nhiều đến các thành phần khác trong hệ thống.

### Kết hợp cả hai mô hình:

Sự kết hợp giữa **Kiến trúc phân tầng** và **MVC** sẽ mang lại các lợi ích sau:

- **Lớp giao diện (View)** sẽ liên kết với lớp **Presentation Layer**, cho phép nhân viên và quản lý thực hiện các thao tác như nhập liệu, tra cứu và báo cáo một cách dễ dàng.
- **Lớp nghiệp vụ (Business Logic)** nằm ở lớp **Controller** của MVC, nơi thực hiện các thao tác quản lý đại lý, lập phiếu xuất hàng, thu tiền và tính toán các thông tin cần thiết.
- **Lớp dữ liệu (Model)** sẽ tương tác với lớp **Data Access Layer** để lưu trữ và quản lý cơ sở dữ liệu.

## Mô hình kiến trúc phần mềm Use Case:



### 3.2 Phần cứng

Phần mềm hoạt động trên các máy tính chạy hệ điều hành Windows 10 trở lên.

# 4 Kế hoạch phát triển

## 4.1 Phân tích yêu cầu

### 4.1.1 Mô tả lại phát biểu bài toán

- Mục tiêu:** Giải quyết vấn đề quản lý thông tin đại lý, đơn hàng, nợ và báo cáo doanh số của công ty TNHH Phân phối Hàng tiêu dùng An Phú bằng một ứng dụng phần mềm. Ứng dụng này sẽ tự động hóa quy trình xử lý dữ liệu, giúp giảm thiểu lỗi và nâng cao hiệu suất.
- Phát biểu bài toán:**
  - "Công ty An Phú cần một ứng dụng quản lý đại lý và đơn hàng để tự động hóa việc lập phiếu xuất hàng, thu tiền, báo cáo doanh số và công nợ, giúp giảm thiểu lỗi và cải thiện dòng tiền."

### 4.1.2 Phân tích yêu cầu chức năng và phi chức năng

- Yêu cầu chức năng:**
  - Quản lý thông tin đại lý: Lưu trữ và cập nhật danh sách đại lý, bao gồm thông tin liên hệ, nợ công, và lịch sử giao dịch.
  - Lập phiếu xuất hàng và thu tiền: Ứng dụng tạo và lưu trữ phiếu xuất hàng và phiếu thu tiền cho mỗi giao dịch.
  - Kiểm tra giới hạn số tiền thu: Hệ thống sẽ kiểm tra để đảm bảo rằng số tiền thu không vượt quá số tiền mà đại lý đang nợ. Nếu số tiền nhập vào vượt quá nợ, hệ thống sẽ đưa ra cảnh báo và yêu cầu điều chỉnh lại số tiền.
  - Báo cáo doanh số và nợ: Tạo báo cáo doanh số hàng tháng/quý/năm và theo dõi nợ công của các đại lý.
  - Chuyển đổi qua lại giữa các chức năng: Người dùng có thể dễ dàng chuyển đổi giữa các chức năng như lập phiếu xuất hàng, báo cáo doanh số, và quản lý nợ.
  - Lưu trữ và in ấn: Cho phép người dùng lưu các biểu mẫu dưới dạng file và in chúng khi cần.

- Chức năng đăng nhập: Hệ thống sẽ yêu cầu đăng nhập khi người dùng truy cập vào ứng dụng.

- **Yêu cầu phi chức năng:**

- Phần mềm phải khởi chạy nhanh và phản hồi nhanh khi thực hiện các chức năng.
- Ứng dụng cần hỗ trợ trên nền tảng Windows 10 trở lên.
- Hệ thống phải có giao diện thân thiện với người dùng, trực quan và dễ sử dụng.
- Bảo mật dữ liệu: Ứng dụng phải đảm bảo rằng thông tin về đại lý và giao dịch được bảo mật, chỉ có nhân viên được ủy quyền mới có thể truy cập.
- Bảo mật mật khẩu: Mật khẩu sẽ được mã hóa trước khi lưu trữ trong cơ sở dữ liệu. Hệ thống hỗ trợ chính sách mật khẩu mạnh, độ dài từ 8 ký tự trở lên, phải có ít nhất một ký tự in hoa và 1 ký tự đặc biệt.

#### 4.1.3 Tìm ra các bên liên quan (stakeholders) và vai trò

- **Mục tiêu:** Xác định các đối tượng liên quan đến ứng dụng và vai trò của họ.
- **Stakeholders:**
  - Ban quản lý công ty An Phú: Người quản lý hệ thống, có quyền truy cập và điều khiển toàn bộ hệ thống.
  - Nhân viên: Người sử dụng phần mềm để quản lý đại lý, quản lý phiếu xuất hàng và thu tiền, theo dõi nợ và báo cáo tài chính.
- **Phần mềm hỗ trợ:** Lucidchart hoặc Microsoft Visio có thể được dùng để vẽ sơ đồ ngữ cảnh mô tả mối quan hệ giữa hệ thống và các bên liên quan.

#### 4.1.4 Tạo ra mô hình trường hợp sử dụng (Use Case)

- **Mục tiêu:** Mô tả các kịch bản chính mà hệ thống tương tác với người dùng.
- **Phần mềm hỗ trợ:** Draw.io để tạo mô hình Use Case.

#### 4.1.5 Tạo ra các mô hình giao diện người dùng

- **Mục tiêu:** Mô hình hóa các giao diện để người dùng hình dung và tương tác với hệ thống.

Ví dụ:

- Giao diện trang chủ: Hiển thị các nút chức năng chính như "Quản lý đại lý", "Lập phiếu xuất hàng", "Thu tiền", "Báo cáo".
- Giao diện lập phiếu xuất hàng: Có các trường nhập thông tin hàng hóa, số lượng, và nút "Tạo phiếu".

- **Framework hỗ trợ:** Bootstrap.

#### 4.1.6 Thành phẩm: Bản báo cáo phân tích yêu cầu

- **Mục tiêu:** Tổng hợp và tạo ra tài liệu yêu cầu rõ ràng để nhóm phát triển sử dụng trong quá trình phát triển phần mềm.
- **Thành phẩm:**
  - Tài liệu mô tả yêu cầu chức năng và phi chức năng chi tiết.
  - Sơ đồ ngữ cảnh và sơ đồ Use Case để minh họa các chức năng chính.
  - Các mẫu giao diện người dùng (mockup) thể hiện các chức năng của ứng dụng.
- **Phần mềm hỗ trợ:** Microsoft Word, Google Docs, hoặc Confluence để viết và quản lý tài liệu phân tích yêu cầu.

## 4.2 Thiết kế phần mềm

### 4.2.1 Tạo mô hình quan niệm

- **Mục tiêu:** Xây dựng mô hình tổng quan của hệ thống, xác định các thành phần chính và mối quan hệ giữa chúng.
- **Nội dung:**
  - Phân tích các thành phần chính như *Đại lý* (*Agency*), *Phiếu xuất hàng* (*Invoice*), *Sản phẩm* (*Product*) và *Thanh toán* (*Payment*).
  - Ví dụ:
    - *Đại lý* (*Agency*): Thông tin chi tiết về đại lý (gồm: tên, địa chỉ, nợ, lịch sử giao dịch...).
    - *Phiếu xuất hàng* (*Invoice*): Mô tả các giao dịch xuất hàng từ công ty đến đại lý (gồm: thông tin sản phẩm, số lượng, giá trị, thời gian giao dịch...).
    - *Sản phẩm* (*Product*): Thông tin về sản phẩm, số lượng còn lại trong kho, và giá bán lẻ/ sỉ...
  - Mối quan hệ giữa các đối tượng này sẽ được xác định để đảm bảo sự tương tác trong hệ thống diễn ra mạch lạc.
- **Thành phần:** Sơ đồ quan niệm tổng quan giúp hình dung rõ hệ thống sẽ bao gồm các thành phần nào và cách chúng tương tác với nhau.

### 4.2.2 Bản thiết kế kiến trúc

- **Mục tiêu:** Lập kế hoạch về cách tổ chức các thành phần của hệ thống và cách các thành phần này giao tiếp với nhau.
- **Nội dung:**
  - Hệ thống sẽ được phân chia thành các mô-đun chính như: *Quản lý đại lý*, *Quản lý sản phẩm*, *Quản lý đơn hàng*, và *Quản lý thanh toán*.
  - Xác định phương thức giao tiếp giữa các mô-đun này và các API (nếu có).
- **Thành phần:** Bản thiết kế kiến trúc mô tả cách thức tổ chức hệ thống phần mềm.

#### 4.2.3 Bản thiết kế dữ liệu

- **Mục tiêu:** Xác định các thông tin cần lưu trữ và cách tổ chức dữ liệu trong hệ thống.
- **Nội dung:**
  - Lập kế hoạch về các dữ liệu quan trọng như thông tin đại lý, sản phẩm, phiếu xuất hàng và thanh toán.
  - Xác định cách các dữ liệu này liên kết với nhau, ví dụ, một đại lý có thể có nhiều phiếu xuất hàng, và các phiếu này có thể chứa nhiều sản phẩm.
- **Thành phẩm:** Thiết kế cơ sở dữ liệu, trong đó mô tả chi tiết các thông tin cần lưu trữ và mối quan hệ giữa các bảng dữ liệu.

#### 4.2.4 Bản thiết kế giao diện người dùng

- **Mục tiêu:** Lên kế hoạch về cách tổ chức giao diện người dùng và trải nghiệm người dùng (UI/UX).
- **Nội dung:**
  - Xác định các màn hình chính của phần mềm, chẳng hạn như: *Trang chủ*, *Lập phiếu xuất hàng*, *Quản lý đại lý*, và *Báo cáo*.
  - Lên kế hoạch về cách các màn hình sẽ kết nối và chuyển tiếp giữa nhau để người dùng có trải nghiệm liền mạch.
- **Thành phẩm:** Mô tả về các màn hình và quy trình luồng công việc của người dùng.

#### 4.2.5 Thành phẩm cuối cùng

- Thành phẩm cuối cùng của quá trình này bao gồm:
  - Mô hình quan niệm, mô tả các thành phần chính và cách chúng liên kết với nhau.
  - Bản thiết kế kiến trúc tổng thể, mô tả tổ chức của hệ thống.
  - Bản thiết kế dữ liệu, xác định các dữ liệu và mối quan hệ giữa chúng.
  - Bản thiết kế giao diện người dùng, mô tả sơ đồ các màn hình chính và luồng điều hướng.

## 4.3 Cài đặt (implement) phần mềm

### 4.3.1 Chia phần mềm thành các module

- **Mục tiêu:** Xác định các module chính của hệ thống và phân chia công việc giữa các lập trình viên để đảm bảo tính phân quyền và hiệu quả trong quá trình phát triển.
- **Các module chính:**
  - Quản lý đại lý (Agency Management): Module này xử lý việc quản lý thông tin của các đại lý, bao gồm thêm mới, chỉnh sửa, và xóa đại lý.
  - Quản lý sản phẩm (Product Management): Đảm nhận việc quản lý thông tin sản phẩm, số lượng hàng tồn kho, giá bán, và cập nhật kho hàng.
  - Lập phiếu xuất hàng (Invoice Management): Xử lý quy trình tạo phiếu xuất hàng, tính toán tổng giá trị đơn hàng, và lưu trữ thông tin về giao dịch.
  - Quản lý thanh toán (Payment Management): Đảm nhận việc ghi nhận thanh toán từ các đại lý, theo dõi nợ, và xử lý báo cáo tài chính.
  - Báo cáo và thống kê (Reporting): Module này xử lý các báo cáo về doanh thu, công nợ, và tình trạng đơn hàng của các đại lý.
- **Phân công công việc:**
  - Mỗi lập trình viên sẽ đảm nhận một hoặc nhiều module tùy thuộc vào quy mô và tính chất của module đó.
  - Các module được phát triển đồng thời nhưng cần phải tuân thủ thiết kế tổng thể và giao tiếp giữa các module được kiểm soát chặt chẽ.

### 4.3.2 Quản lý và tích hợp mã nguồn

- **Mục tiêu:** Đảm bảo quá trình phát triển diễn ra đồng bộ, tránh xung đột mã nguồn và dễ dàng tích hợp giữa các module.
- **Công cụ:** Sử dụng hệ thống quản lý mã nguồn như *Git* (*GitHub*, *GitLab*) để theo dõi và kiểm soát các thay đổi trong quá trình phát triển. Mỗi lập trình viên sẽ làm việc trên nhánh riêng của module mình, sau đó tích hợp lại trong nhánh chính sau khi đã hoàn thành và kiểm thử module.
- **Lên lịch tích hợp:** Xác định thời gian tích hợp thường xuyên để phát hiện sớm lỗi liên quan đến tương tác giữa các module.

#### 4.3.3 Kiểm tra từng module

- **Mục tiêu:** Đảm bảo rằng mỗi module hoạt động độc lập và đáp ứng đúng yêu cầu trước khi tích hợp toàn hệ thống.
- **Quy trình kiểm tra:**
  - Mỗi module sẽ được kiểm thử cục bộ bởi lập trình viên phụ trách, sau đó được kiểm thử tích hợp để đảm bảo tính tương thích với các module khác.
  - Các công cụ kiểm thử đơn vị (*unit testing*) và kiểm thử tích hợp sẽ được áp dụng để phát hiện lỗi sớm.

#### 4.3.4 Thành phẩm

- **Mục tiêu:** Sau quá trình cài đặt, hệ thống phải đáp ứng đầy đủ các yêu cầu đã phân tích và thiết kế.
- **Thành phẩm:**
  - Hệ thống phần mềm hoàn chỉnh với các chức năng như quản lý đại lý, lập phiếu xuất hàng, quản lý thanh toán, và báo cáo.
  - Phần mềm hoạt động trơn tru, không có lỗi lớn và được kiểm thử đầy đủ theo tiêu chuẩn.

## 4.4 Kiểm thử phần mềm

### 4.4.1 Lập kế hoạch kiểm thử

- **Mục tiêu:** Xây dựng kế hoạch kiểm thử phần mềm toàn diện để đảm bảo phần mềm hoạt động đúng theo yêu cầu và không có lỗi nghiêm trọng.
- **Nội dung:**
  - Lựa chọn kỹ thuật kiểm thử:
    - Sử dụng *kiểm thử hộp đen (Black-box testing)* để kiểm tra các chức năng mà không cần quan tâm đến mã nguồn, đảm bảo phần mềm đáp ứng đúng yêu cầu.
    - *Kiểm thử hộp trắng (White-box testing)* được áp dụng cho kiểm thử chi tiết mã nguồn trong một số trường hợp cần thiết, đặc biệt là kiểm thử đơn vị (*unit testing*).
  - Kiểm thử chức năng và phi chức năng:
    - Kiểm thử chức năng bao gồm các kiểm tra trên từng module như *quản lý đại lý*, *lập phiếu xuất hàng*, *quản lý thanh toán*, và *báo cáo*.
    - Kiểm thử phi chức năng sẽ tập trung vào hiệu suất, tính tương thích, và khả năng bảo mật.
  - Đối tượng kiểm thử:
    - Kiểm thử sẽ được thực hiện trên các module đã phát triển xong và tích hợp. Đối tượng bao gồm cả chức năng của hệ thống và tài liệu hướng dẫn người dùng, nếu có.

### 4.4.2 Tạo danh sách các test case

- **Mục tiêu:** Xây dựng hệ thống các test case để đảm bảo rằng tất cả các chức năng của hệ thống được kiểm tra đầy đủ.
- **Nội dung:**
  - Thiết lập các test case chi tiết: Mỗi test case cần xác định rõ:
    - Input (đầu vào): Điều kiện hoặc dữ liệu cần nhập vào.
    - Expected output (kết quả mong đợi): Kết quả đầu ra dự kiến khi thực hiện kiểm thử.

- Actual output (kết quả thực tế): Kết quả thực tế nhận được khi thực hiện kiểm thử.
- Status (trạng thái): Thành công hay thất bại (pass/fail) dựa trên sự so sánh giữa kết quả mong đợi và kết quả thực tế.

#### 4.4.3 Thực hiện kiểm thử

- **Mục tiêu:** Thực hiện kiểm thử dựa trên các test case đã xây dựng để đảm bảo chất lượng phần mềm.
- **Nội dung:**
  - Các lập trình viên và đội ngũ kiểm thử viên sẽ lần lượt thực hiện từng test case theo kế hoạch đã đề ra.
  - Mọi lỗi phát sinh sẽ được ghi lại và báo cáo để tiến hành sửa lỗi (*debugging*) và kiểm thử lại cho đến khi phần mềm đạt yêu cầu.

#### 4.4.4 Thành phẩm

- **Mục tiêu:** Tạo ra các tài liệu liên quan đến quá trình kiểm thử, nhằm đảm bảo rằng tất cả các chức năng đã được kiểm tra kỹ lưỡng.
- **Thành phẩm:**
  - Kế hoạch kiểm thử phần mềm: Tài liệu chi tiết về phương pháp, kỹ thuật và đối tượng kiểm thử.
  - Danh sách các test case: Hệ thống các test case chi tiết cho từng chức năng và module của phần mềm.
  - Báo cáo kết quả kiểm thử: Kết quả của mỗi test case sau khi kiểm thử, bao gồm cả thông tin về những lỗi đã gặp phải và cách xử lý chúng.

## 4.5 Triển khai, bảo trì

### 4.5.1 Triển khai phần mềm

- **Mục tiêu:** Đưa phần mềm vào sử dụng chính thức trong môi trường thực tế của Công ty TNHH Phân phối Hàng tiêu dùng An Phú.
- **Nội dung:**
  - Cài đặt và cấu hình: Cài đặt phần mềm trên hệ thống của công ty, đảm bảo tất cả các máy tính sử dụng phần mềm đều được cấu hình đúng cách.
  - Hướng dẫn sử dụng: Tạo tài liệu hướng dẫn và tổ chức buổi đào tạo cho nhân viên để họ có thể sử dụng phần mềm một cách hiệu quả. Buổi đào tạo sẽ hướng dẫn các chức năng chính như quản lý đại lý, lập phiếu xuất hàng, và theo dõi công nợ.
  - Kiểm tra hoạt động sau triển khai: Kiểm tra việc phần mềm hoạt động ổn định sau khi được cài đặt và sử dụng thực tế, đảm bảo rằng không có vấn đề nào phát sinh trong quá trình triển khai.
- **Thành phẩm:**
  - Phần mềm được cài đặt và hoạt động trên hệ thống của công ty.
  - Tài liệu hướng dẫn sử dụng và đào tạo nhân viên.

### 4.5.2 Nhận phản hồi từ người dùng

- **Mục tiêu:** Thu thập thông tin phản hồi từ nhân viên và người dùng phần mềm để xác định các vấn đề phát sinh hoặc đề xuất cải tiến.
- **Nội dung:**
  - Thu thập phản hồi: Thông qua các cuộc khảo sát, phỏng vấn hoặc trao đổi trực tiếp với người dùng để hiểu rõ trải nghiệm của họ với phần mềm.
  - Đánh giá phản hồi: Phân tích các phản hồi nhận được để xác định xem phần mềm có đáp ứng được các yêu cầu kinh doanh, hoặc nếu có vấn đề kỹ thuật hoặc thiếu sót trong chức năng.

#### 4.5.3 Bảo trì và nâng cấp

- **Mục tiêu:** Bảo trì phần mềm để đảm bảo hoạt động ổn định lâu dài, đồng thời cải tiến và nâng cấp dựa trên phản hồi của người dùng.
- **Nội dung:**
  - Sửa lỗi (nếu có): Nếu phần mềm phát sinh lỗi trong quá trình sử dụng, tiến hành phân tích và quay lại bước 4.1 Kiểm thử để kiểm tra và sửa lỗi.
  - Nâng cấp tính năng: Nếu cần thêm tính năng mới hoặc cải tiến tính năng hiện tại theo nhu cầu của công ty, phần mềm sẽ được nâng cấp và triển khai lại.
  - Bảo trì định kỳ: Thực hiện bảo trì định kỳ nhằm đảm bảo phần mềm hoạt động ổn định, tối ưu hóa hiệu suất và đáp ứng nhu cầu thay đổi của công ty trong tương lai.
- **Thành phẩm:**
  - Phần mềm sau khi bảo trì và nâng cấp hoạt động ổn định hơn, cải tiến theo yêu cầu của người dùng.

# 5

## Kế hoạch nhân sự & chi phí

- **Nhóm dự án gồm 5 thành viên, mỗi thành viên đảm nhiệm các vị trí tương ứng:**

- 22120289 – Hồ Ngọc Trung Quân: Designer, Developer
- 22120304 – Nguyễn Thị Kim Quý: Developer, Tester
- 22120338 – Đỗ Hạnh Thảo: Business Analyst, Developer
- 22120341 – Nguyễn Minh Thiện: Developer, Tester
- 22120351 – Vũ Thị Kim Thoa: Project Manager, Developer

- **Chi phí về thời gian cho dự án:**

- Phân tích yêu cầu: Ước tính thời gian khoảng 20h
- Thiết kế phần mềm: Ước tính thời gian khoảng 50h
- Cài đặt phần mềm: Ước tính thời gian khoảng 125h
- Kiểm thử phần mềm: Ước tính thời gian khoảng 10h
- Triển khai, bảo trì: Ước tính thời gian khoảng 5h cho triển khai, thời gian dài hạn cho việc bảo trì (nếu có).

Tổng thời gian cho dự án dự kiến khoảng **210h làm việc liên tục** (chưa tính thời gian bảo trì).

Thời gian hoàn thành dự án: **8 tuần**

- Tuần 1 – tuần 7: làm việc 30h / 1 tuần
- Tuần 8: bảo trì, sửa lỗi, hoàn thiện sản phẩm

- **Chi phí về ngân sách cho dự án:**

- Thiết bị phần cứng cài đặt / triển khai phần mềm: 0 VND
- Chi phí cho nhân sự: 50.000 VND / 1h làm việc

Như vậy tổng ngân sách cho dự án dự kiến dựa theo số giờ làm việc là **10.500.000 VND**.