 

PROGRAM TITLE: BTEC in Computing (Software Engineering)

UNIT TITLE: Software Development Lifecycles

ASSIGNMENT NUMBER: 1

ASSIGNMENT NAME: Houseplant store management system

SUBMISSION DATE: 31/10/2024

DATE RECEIVED: 31/10/2024

TUTORIAL LECTURER: Nguyen Van Tuyen

WORD COUNT:

STUDENT NAME: TRƯƠNG VĂN MINH

STUDENT ID: BKC15067

MOBILE NUMBER: 0828928569

|  |
| --- |
| **Summative Feedback:**  You achieved:  **Strength:**  **Weakness:**  **How to improve:**  Grade:  Date:  Assessor: |
| **Internal verification:**  **You achieved:**  **Strength:**  •  **Weakness:**  •  **How to improve:**  •  **Grade:**  **Date:**  **Assessor:** |
|  |

1. **Mô tả vòng đời phát triển phần mềm**
2. **Vòng đời phát triển phần mềm (SDLC) (P1)**

* Vòng đời phát triển phần mềm (SDLC) là một quy trình giúp các nhóm phát triển thiết kế và xây dựng phần mềm chất lượng cao, tiết kiệm chi phí và thời gian.
* Các giai đoạn chính trong SDLC:
  + Thu thập và phân tích yêu cầu (Requirement Analysis)
    - Giai đoạn này là khi chúng ta tìm hiểu câu trả lời cho câu hỏi “Các vấn đề hiện tại là gì?” Đây là giai đoạn quan trọng nhất trong SDLC, vì nó xác định những gì chúng ta sẽ phát triển. Nếu bạn đang phát triển cho khách hàng, giai đoạn này có thể bao gồm cuộc họp để thảo luận về nhu cầu, mục tiêu và kỳ vọng của họ.
  + Lập kế hoạch (Planning):
    - Tại giai đoạn này, chúng ta xác định lịch trình, tài nguyên, và các bước cần thiết để phát triển phần mềm. Lập kế hoạch giúp đảm bảo rằng dự án được thực hiện một cách có hệ thống và hiệu quả.
  + Thiết kế (Design):
    - Giai đoạn này liên quan đến việc tạo ra thiết kế chi tiết cho phần mềm. Chúng ta xác định cấu trúc, giao diện người dùng, và các thành phần khác của hệ thống.
  + Mã hóa (Coding)
    - Tại giai đoạn này, chúng ta viết mã để triển khai thiết kế đã được xác định. Đây là giai đoạn mà phần mềm thực sự được tạo ra
  + Kiếm thử (Testing):
    - Chúng ta kiểm tra phần mềm để đảm bảo rằng nó hoạt động đúng và đáp ứng các yêu cầu đã đề ra. Kiểm thử bao gồm kiểm tra chức năng, hiệu suất, và bảo mật.
  + Triển khai (Deployment)
    - Giai đoạn này liên quan đến việc triển khai phần mềm vào môi trường sản xuất. Chúng ta đưa phần mềm vào sử dụng thực tế.
  + Bảo trì (Maintenance):
    - Sau khi phần mềm đã được triển khai, chúng ta tiếp tục duy trì và cải thiện nó theo thời gian
* Hai mô hình vòng đời tuần tự: Mô hình thác nước (Waterfull), Mô hình chữ V (V-Model)

1. Mô hình thác nước (Waterfall)
   1. Tổng quan:
      * Mô hình thác nước là một trong những mô hình quản lý dự án được ưa chuộng
      * Các giai đoạn trong mô hình này diễn ra lần lượt và nối tiếp nhau.
   2. Các giai đoạn
      * Phân tích và mô tả yêu cầu
      * Thiết kế
      * Thực hiện và thử nghiệm
      * Kiểm tra tích hợp
      * Vận hành và bảo trì
   3. Ưu điểm, nhược điểm:
      * Ưu điểm
        + Dễ sử dụng và dễ hiểu
        + Rà soát chất lượng ở từng khâu
      * Nhược điểm
        + Khó thích nghi với thay đổi
        + Không phản ánh tốt sự phức tạp của dự án
2. Mô hình chữ V (V-Model)
   1. Tổng quan
      * Mô hình chữ V là một biến thể của mô hình thác nước
      * Mô hình này tích hợp chặt chẽ hoạt động kiểm thử vào từng giai đoạn phát triển, tạo nên một quy trình song hành đồng bộ và hiệu quả
   2. Các giai đoạn
      * Thu thập và phân tích yêu cầu (Requirementt Analysis)
      * Thiết kế (Design)
      * Phát triển (Development)
      * Kiểm thử (Testing)
   3. Ưu điểm, nhược điểm
      * Ưu điểm
        + Tích hợp kiểm thử sớm và liên tục
        + Giảm thiểu rủi ro và tối ưu hóa quy trình kiểm thử
      * Nhược điểm
        + Khó thích nghi với thay đổi
        + Cần sự chú ý và quản lý cẩn thận để đảm bảo đồng bộ giữa phát triển và kiểm thử

* Hai mô hình vòng đời lặp:

1. Mô hình xoắn ốc (Spiral Model)
2. Tổng quan:
   * + Mô hình xoắn ốc là một biến thể của mô hình thác nước
     + Nó kết hợp các yếu tố của mô hình thác nước và mô hình phát triển lặp, tập trung vào việc quản lý rủi ro thông qua các chu kỳ lặp lại và cải tiến liên tục
3. Các giai đoạn của mô hình:
   * + Lập kế hoạch (Planning phase)
     + Phân tích rủi ro (Risk analysis phase)
     + Thực thi kỹ thuật (Engineering phase)
     + Đánh giá (Evaluation phase)
4. Ưu điểm, Nhược điểm:
   * + Ưu điểm:
       - Phân tích rủi ro cao
       - Ước tính chi phí dễ dàng
     + Nhược điểm:
       - Khó thích nghi với thay đổi
       - Đòi hỏi kiến thức và kỹ năng quản lý cao h
5. Mô hình Agile (Agile Model)
   * + 1. Tổng quan:
     + Mô hình Agile là một cách lặp đi lặp lại để quản lý dự án, tập trung vào việc chia nhỏ các dự án lớn thành các nhiệm vụ dễ quản lý hơn
       1. Các giai đoạn chính:
     + Lập kế hoạch (Planning)
     + Phân tích (Analysis)
       1. Ưu điểm và nhược điểm:
     + Ưu điểm:
       - Tính linh hoạt
       - Tập trung vào giá trị thực sự
     + Nhược điểm:
       - Yêu cầu sự hợp tác cao
       - Khó thích nghi với dự án lớn

2. **Quản lý rủi ro trong các mô hình** (P2)

a. Rủi ro trong SDLC là gì?

- Là một vấn đề tiềm năng có thể gây ảnh hưởng đến sự thành công của dự án phát triển

- Trong SDLC, rủi ro có thể xuất hiện từ nhiều nguồn, bao gồm thay đổi yêu cầu, vấn đề kỹ thuật, thiếu tài nguyên, không hiểu rõ yêu cầu của khách hàng.

b. Quản lý rủi ro của các mô hình là gì?

- Quản lý rủi ro tập trung vào việc xác định, đánh giá và ứng phó với các yếu tố rủi ro. Nó giúp giảm thiểu tác động tiêu cực của rủi ro và tận dụng cơ hội từ các biến động tích cực.

c. Cách quản lý rủi ro trong một số mô hình SDLC

- Mô hình xoắn ốc (Spiral Model):

+ Mô hình xoắn ốc kết hợp các chu trình nhỏ lặp lại của mô hình lặp đi lặp lại với quy trình tuần tự tuyến tính của mô hình thác nước.

+ Trong giai đoạn phân tích rủi ro, quá trình phân tích được thực hiện để xác định rủi ro và đưa ra các giải pháp thay thế

- Mô hình thác nước (Waterfall Model)

+ **Xác định rủi ro sớm:** Trong mô hình Waterfall, các giai đoạn phát triển diễn ra tuần tự, do đó rủi ro cần được xác định càng sớm càng tốt trong giai đoạn yêu cầu và thiết kế.

+ **Lập kế hoạch dự phòng:** Do đặc thù của mô hình Waterfall là rất khó thay đổi sau khi đã hoàn thành giai đoạn nào đó, cần phải có kế hoạch dự phòng và các tài liệu kỹ thuật chi tiết để giải quyết rủi ro tiềm ẩn.

+ **Kiểm tra và đánh giá liên tục:** Mặc dù không phải là mô hình linh hoạt, nhưng cần có các buổi kiểm tra định kỳ và đánh giá tình hình dự án để đảm bảo rủi ro được phát hiện và xử lý kịp thời.

1. **Ví dụ và giải thích tại sao cần lựa chọn mô hình phát triển phù hợp với dự án(M1)**

* Đưa ra một dự án ví dụ
  + Houseplant store management system (Hệ thống quản lý cửa hàng cây xanh)
* Lựa chọn mô hình phát triển cho dự án trên:
  + Mô hình Agile
* Giải thích lý do lựa chọn mô hình đó
  + Linh hoạt với thay đổi yêu cầu, giúp hệ thống dễ dàng thích nghi khi có sự thay đổi trong quá trình phát triển.
  + Phát hành sớm và liên tục, giúp cửa hàng có thể sử dụng hệ thống ngay từ những phiên bản đầu
  + Phản hồi thường xuyên từ khách hàng, đảm bảo hệ thống luôn phù hợp với nhu cầu thực tế.
  + Giảm thiểu rủi ro, nhờ việc phát hiện và xử lý lỗi sớm.
  + Tập trung vào tính năng cốt lõi trước, cải thiện hiệu quả cho cửa hàng

1. **Nghiên cứu khả thi**

**1. Báo cáo khả thi:** (P3)

- Báo cáo khả thi là một tài liệu phân tích và đánh giá tính khả thi của một dự án hoặc đề xuất. Nó xem xét các yếu tố như kỹ thuật, kinh tế, pháp lý, thời gian, và tài chính để xác định xem liệu dự án có thể thực hiện được hay không, và nếu có thì dự án có đáng đầu tư và phát triển không. Báo cáo khả thi thường được thực hiện trong giai đoạn đầu của một dự án để giúp các bên liên quan đưa ra quyết định sáng suốt về việc có nên tiến hành dự án hay không.

- Mục đích của báo cao khả thi:

* Đánh giá tính khả thi của dự án: Xem xét liệu dự án có khả thi về mặt kỹ thuật, tài chính, và thời gian hay không. Báo cáo giúp xác định các rủi ro tiềm ẩn và các yếu tố có thể ảnh hưởng đến sự thành công của dự án.
* Hỗ trợ ra quyết định: Báo cáo khả thi cung cấp thông tin cần thiết để các nhà quản lý, nhà đầu tư, và các bên liên quan khác có thể đưa ra quyết định có nên tiếp tục đầu tư hoặc phát triển dự án không. Nó giúp xác định dự án có khả năng mang lại lợi ích và giá trị lớn hơn so với chi phí và rủi ro không.
* Xác định yêu cầu nguồn lực: Giúp xác định các nguồn lực cần thiết cho dự án, bao gồm tài chính, nhân lực, thiết bị, và thời gian. Điều này giúp đảm bảo rằng các nguồn lực có sẵn và được phân bổ một cách hợp lý.
* Đưa ra kế hoạch hành động: Báo cáo khả thi không chỉ đánh giá tính khả thi của dự án mà còn đưa ra các khuyến nghị và kế hoạch hành động để thực hiện dự án một cách hiệu quả nếu dự án được chấp thuận.
* Giảm thiểu rủi ro: Bằng cách xác định và đánh giá các rủi ro tiềm ẩn trước khi dự án bắt đầu, báo cáo khả thi giúp giảm thiểu nguy cơ thất bại và đảm bảo rằng dự án được chuẩn bị tốt trước khi triển khai.

1. **Làm thế nào để so sánh các giải pháp công nghệ có thể áp dụng vào dự án** (P4):

* **B1:** Xác định yêu cầu và mục tiêu của dự án: Hiểu rõ các yêu cầu chức năng, phi chức năng, và mục tiêu của dự án sẽ giúp bạn chọn giải pháp công nghệ phù hợp
* **B2:** Danh sách các giải pháp công nghệ tiềm năng: Tạo danh sách các công nghệ, nền tảng, hoặc công cụ có thể đáp ứng yêu cầu của dự án.
* **B3:** Đánh giá từng giải pháp theo các tiêu chí
  + Tính khả thi: Giải pháp có phù hợp với yêu cầu kỹ thuật và khả năng hiện tại của bạn không?
  + Chi phí: So sánh chi phí triển khai và bảo trì của từng giải pháp.
  + Hiệu suất: Giải pháp có đáp ứng yêu cầu về hiệu suất và khả năng mở rộng không?
  + Bảo mật: Đánh giá mức độ bảo mật mà giải pháp cung cấp.
  + Hỗ trợ và cộng đồng: Giải pháp có cộng đồng người dùng và hỗ trợ kỹ thuật tốt không?
  + Khả năng tích hợp: Giải pháp có dễ dàng tích hợp với các hệ thống hiện tại hoặc các công nghệ khác không?
* **B4:** Thử nghiệm và so sánh: Nếu có thể, thực hiện các thử nghiệm hoặc xây dựng nguyên mẫu để so sánh hiệu quả thực tế của các giải pháp.
* **B5:** Đánh giá rủi ro và lợi ích: Xem xét các rủi ro liên quan đến từng giải pháp và cân nhắc lợi ích lâu dài.
* **B6:** Ra quyết định: Dựa trên các tiêu chí đánh giá và kết quả thử nghiệm, chọn giải pháp công nghệ phù hợp nhất với yêu cầu của dự án.

**3. Các thành phần của một báo cáo khả thi (M2):**

**-** Một báo cáo khả thi gồm các thành phần:

+ Tóm tắt điều hành (Executive Summary)

. Tổng quan về dự án, mục tiêu chính và kết luận của báo cáo. Phần này cung cấp cái nhìn nhanh về các khía cạnh quan trọng nhất.

+ Mô tả dự án (Project Description)

. Giới thiệu chi tiết về dự án, bao gồm mục tiêu, phạm vi, bối cảnh thực hiện và các yêu cầu cơ bản.

+ Nghiên cứu thị trường (Market Research)

. Phân tích thị trường mục tiêu, nhu cầu, đối thủ cạnh tranh, và xu hướng thị trường. Phần này giúp xác định tính khả thi về thị trường của dự án

+ Phân tích kỹ thuật (Technical Feasibility)

. Đánh giá các yêu cầu kỹ thuật của dự án, bao gồm công nghệ, thiết bị, phần mềm, và nhân lực cần thiết để thực hiện.

+ Phân tích tài chính (Financial Feasibility)

. Bao gồm chi phí ước tính, nguồn vốn, lợi nhuận dự kiến và phân tích lợi ích – chi phí để đánh giá khả năng tài chính của dự án.

+ Phân tích tổ chức và nhân sự (Organizational and Staffing Feasibility)

. Đánh giá cơ cấu tổ chức, nguồn lực nhân sự cần thiết để quản lý và thực hiện dự án.

+ Phân tích pháp lý (Legal Feasibility)

. Kiểm tra các khía cạnh pháp lý liên quan đến dự án như giấy phép, luật pháp và quy định cần tuân thủ

+ Đánh giá rủi ro (Risk Assessment)

. Xác định các rủi ro tiềm ẩn và đề xuất biện pháp giảm thiểu.

+ Phân tích tác động môi trường (Environmental Impact Assessment)

. Đánh giá tác động của dự án đối với môi trường, nếu có, và các biện pháp giảm thiểu tác động tiêu cực.

+ Kết luận và kiếm nghị (Conclusion and Recommendations)

. Tóm tắt các phát hiện chính và đưa ra đề xuất về việc tiếp tục hay không triển khai dự án

**3. Theo dõi các yêu cầu phần mềm: (M3)**

**-** Có cách giải pháp nào để quản lý các phiên bản của phần mềm?

+ Sử dụng hệ thống kiểm soát phiên bản (Version Control System – VCS)

+ Sử dụng GitFlow hoặc GitHub Flow

+ Sử dụng Phần mềm Quản lý Phiên bản

+ Gắn thẻ (Tagging) và phát hành (Releases)

+ Quản lý bản phát hành (Release Management)

+ Kiếm soát và Tự động Hóa Phiên bản (CI/CD)

- Trình bày Schedule Baseline của dự án.

1. Khởi động dự án (Project Initiation)

- Thời gian: 1 tuần

- Nhiệm vụ:

+ Xác định mục tiêu và phạm vi dự án

+ Lập danh sách các bên liên quan

- Kết quả: Kế hoạch dự án ban đầu được phê duyệt.

2. Phân tích yêu cầu

- Thời gian: 2 tuần

- Nhiệm vụ  
 + Thu thập yêu cầu từ các bên liên quan

+ Xác định tính năng cần thiết

+ Tạo tài liệu yêu cầu chức năng

- Kết quả: Tài liệu yêu cầu được phê duyệt.

3. Thiết kế hệ thống

- Thời gian: 4 tuần

- Nhiệm vụ:

- Thiết kế sơ đồ hệ thống

- Thiết kế giao diện người dùng

- Xây dựng cơ sở dữ liệu

4. Phát triển hệ thống:

- Thời gian: khoảng 2 tháng

- Nhiệm vụ

+ Tạo chức năng

+ Xây dựng giao diện cửa hàng và bảng quản lý

+ Tích hợp cơ sở dự liệu

5. Kiếm thử hệ thống:

- Thời gian: 3 tuần

- Nhiệm vụ:

+ Kiếm thử đơn vị

+ Kiếm thử người dùng

+ Kiếm thử tích hợp

6. Triển khai

- Thời gian: 2 tuần

- Nhiệm vụ

+ Cài đặt hệ thống

+ Đào tạo người dùng

+ Chuẩn bị tài liệu hướng dẫn sử dụng

7. Bảo trì và hỗ trợ

**4. Trình bày về 2 cách tiếp cận cải thiện chất lượng phần mềm. (M4)**

**-** Tổng quan về cách tiếp cận.

+ Phương pháp luận (Methodology)

.  Waterfall: Là phương pháp luận tuyến tính, từng bước đi theo trình tự: từ phân tích yêu cầu đến triển khai và bảo trì. Waterfall phù hợp nếu yêu cầu dự án rõ ràng ngay từ đầu và ít thay đổi trong suốt quá trình phát triển.

 Agile: Là phương pháp tiếp cận lặp đi lặp lại với các chu kỳ ngắn (sprint) giúp dễ dàng thích ứng với các thay đổi yêu cầu. Phương pháp này phù hợp nếu dự án có nhiều yêu cầu thay đổi hoặc cần phản hồi nhanh từ khách hàng.

+ Giai đoạn chính (Key Phases)

 Khởi động (Initiation**)**: Xác định mục tiêu, phạm vi và cấu trúc nhóm dự án.

 Lập kế hoạch (Planning): Xác định các hoạt động chi tiết, tiến độ, nguồn lực, và quản lý rủi ro.

 Phân tích yêu cầu (Requirements Analysis): Thu thập và phân tích yêu cầu từ khách hàng và các bên liên quan.

 Thiết kế hệ thống (System Design): Thiết kế chi tiết phần mềm bao gồm kiến trúc, cơ sở dữ liệu, và giao diện người dùng.

 Phát triển (Development): Tiến hành lập trình và tích hợp các thành phần hệ thống.

 Kiểm thử (Testing): Kiểm thử phần mềm để đảm bảo chất lượng và đáp ứng đúng yêu cầu.

 Triển khai (Deployment): Đưa phần mềm vào sử dụng thực tế.

 Bảo trì và hỗ trợ (Maintenance & Support): Giải quyết các vấn đề sau khi triển khai và nâng cấp hệ thống khi cần thiết.

**III. Thực hiện một cuộc điều tra phần mềm:**

**1. Thực hiện một cuộc điều tra phần mềm (P5)**

**Dự án:** Houseplant store management system (Hệ thống quản lý cửa hàng cây xanh)

- Trình bày về dự án đã chọn:

+  Một cửa hàng bán cây xanh muốn tạo 1 website chuyên loại cây cảnh, cây trang trí của nhiều mẫu mã, chủ đề khác nhau. Cửa hàng muốn kinh doanh bằng hình thức bán hàng online. Khách hàng khi muốn mua hàng có thể chọn các mẫu mã khác nhau của sản phẩm. Khách hàng cũng có thể tìm kiếm các sản phẩm trong cùng một loại và có thể lọc các sản phẩm theo khoảng giá khác nhau, các màu khác nhau của một sản phẩm có thể có số lượng khác nhau nhưng giá bằng nhau Đối với bán hàng online, khách hàng cần truy cập vào website của hàng, sau đó chọn mặt hàng muốn mua, thêm vào giỏ hàng rồi thanh toán. Khách hàng mua hàng online cần phải tạo tài khoản của mình, đăng nhập thì mới có thể đặt được đơn hàng. Khách hàng cũng có thể mua nhiều sản phẩm khác nhau hoặc cũng một sản phẩm nhưng với số lượng nhiều hơn. Sau khi đặt đơn hàng, khách hàng có thể xem lại lịch sử đơn hàng đã đặt và có thể hủy đơn hàng đã đặt trong trường hợp cửa hàng chưa chấp nhận đơn đặt hàng. Khi có khách hàng đặt hàng online, chủ cửa hàng cần xác nhận đơn hàng. Một đơn hàng sẽ có một trong các trạng thái sau: chờ xác nhận, đã xác nhận, đã hoàn thành và đã hủy Chủ cửa hàng cần xem được ai là người xác nhận đơn hàng, tổng doanh thu của cửa hàng trong từng tháng (các ngày trong một tháng) hoặc theo năm (các tháng trong một năm), xem được các sản phẩm nào đang gần hết hàng, các sản phẩm đang bán chạy trong cửa hàng (bán nhiều nhất trong vòng 1 tháng).

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |

- Các yêu cầu chức năng của dự án:

+ Admin:

+ Quản lý nhân viên bán hàng (Thêm, sửa, xóa)

+ Quản lý khách hàng

+ Quản lý sản phẩm(Thêm, sửa, xóa)

+ Quản lý đơn hàng(Xem đơn hàng của khách hàng)

+ Khách hàng:

+Xem sản phẩm

+Tìm kiếm sản phẩm

+ Đặt hàng

+Truy cập giỏ hàng

* Các yêu cầu phi chức năng:
  + Bảo mật
  + Độ tin cậy
  + Khả năng bảo trì
  + Hiệu suất
  + Giao diện người dùng
* Tính khả thi của dự án
  + Về công nghệ
    - Sử dụng MySQL, php, composer, PHP storm, bootstrap 5,
  + Về chi phí
    - Thuê lập trình viên, designer trong 2 tháng sẽ là khoảng 10tr/người
    - Chi phí bảo trì: 10 triệu
    - Chi phí quảng cáo: 10 triệu
  + Về thời gian và nguồn lực
    - Thời gian
      * Phân tích và lập kế hoạch(2 – 4 tuần)
      * Thiết kế(4 – 5 tuần)
      * Phát triển (Hơn 1 tháng)
      * Kiếm thử (3 – 5 tuần)
      * Triển khai (2 – 4 tuần)
      * Bảo trì (Liên tục)
    - Nguồn lực:
      * Quản lý dự án
      * Lập trình viên
      * Quản lý cơ sở dữ liệu
      * Desginer
      * Nhân viên hỗ trợ

**2. Sử dụng các công cụ / kỹ thuật phân tích phần mềm thích hợp để thực hiện điều tra phần mềm (P6):**

**-** Phần mềm có các bên liên quan nào?

+ Chủ cửa hàng và quản lý cửa hàng

+ Nhân viên bán hàng

+ Khách hàng

+ Nhà cung cấp

+ Nhân viên quản lý hệ thống

+ Nhà phát triển

- Sơ đồ luồng dữ liệu

+ Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) là một công cụ dùng để biểu diễn dòng chảy của dữ liệu trong một hệ thống thông tin. DFD mô tả cách thức dữ liệu được nhập vào, xử lý và lưu trữ trong hệ thống, cũng như cách dữ liệu được truyền giữa các thành phần khác nhau của hệ thống.

+ Sơ đồ luồng dữ liệu gồm:

* Sơ đồ luồng dữ liệu cấp 0 (DFD level 0) – Sơ đồ bối cảnh (Context Diagram)
* Sơ đồ luồng dữ liệu cấp 1 (DFD level 1)
* Sơ đồ luồng dữ liệu cấp 2 (DFD level 2) và các cấp chi tiết hơn
* Sơ đồ luồng dữ liệu 0 của dự án:
  + **Quy trình chính (Process)**
    - Hệ thống quản lý cửa hàng cây xanh được biểu diễn dưới dạng một quy trình chính duy nhất trong DFD mức 0. Quy trình này sẽ xử lý tất cả các hoạt động của cửa hàng như quản lý đơn hàng, quản lý tồn kho, và thông tin khách hàng.
* **Thực thể bên ngoài (External Entities)**
* **Khách hàng (Customer):** Thực thể này gửi yêu cầu mua hàng đến hệ thống và nhận thông tin đơn hàng hoặc hóa đơn từ hệ thống.
* **Nhân viên bán hàng (Sales Staff):** Nhập thông tin đơn hàng vào hệ thống và nhận thông tin về hàng tồn kho, khách hàng, và hóa đơn.
* **Nhà cung cấp (Supplier):** Cung cấp thông tin về hàng hóa mới hoặc cập nhật tình trạng hàng hóa. Nhận đơn đặt hàng từ hệ thống.
* **Quản lý cửa hàng (Store Manager):** Xem các báo cáo và số liệu thống kê từ hệ thống.
* **Luồng dữ liệu (Data Flows)**
  + Đơn hàng từ Khách hàng đến Houseplant Store Management System (Phần mềm quản lý cửa hàng cây xanh).
  + Thông tin đơn hàng/hóa đơn từ hệ thống đến Khách hàng.
  + Yêu cầu đặt hàng mới từ Nhân viên bán hàng đến hệ thống.
  + Thông tin hàng tồn kho, khách hàng, hóa đơn từ hệ thống đến Nhân viên bán hàng.
  + Đơn đặt hàng mới từ hệ thống đến Nhà cung cấp.
  + Cập nhật hàng hóa và tình trạng đơn hàng từ Nhà cung cấp đến hệ thống.
  + Yêu cầu báo cáo và số liệu thống kê từ Quản lý cửa hàng đến hệ thống.
  + Báo cáo doanh thu và tồn kho từ hệ thống đến Quản lý cửa hàng.
* Sơ đồ lường dữ liệu 1:
  + Các thực thể bên ngoài (External Entities):
    - Khách hàng: Đặt hàng, nhận hàng, và thanh toán.
    - Quản lý: Cập nhật thông tin sản phẩm, quản lý đơn hàng, và xem báo cáo
    - Nhà cung cấp: Cung cấp cây xanh và vật tư cho cửa hàng.
* Các quy trình chính (Processes)
  + - Quản lý sản phẩm: thêm, sửa, xóa thông tin cây xanh và lưu trữ thông tin
    - Xử lý đơn hàng
    - Thanh toán và giao dịch
    - Các kho dữ liệu (Data Stores)
      * Kho sản phẩm
      * Cơ sở dữ liệu đơn hàng
* Sơ đồ ERD của phần mềm.
  + Các thực thể
    - Khách hàng
      * Mã khách hàng
      * Tên khách hàng
      * Số điện thoại
      * Địa chỉ
      * Email
    - Sản phẩm
      * Mã sản phẩm,
      * Tên sản phẩm
      * Giá
      * Số lượng
      * Mô tả
    - Đơn hàng
      * Mã đơn hàng
      * Ngày đặt hàng
      * Tổng tiền
      * Trạng thái
    - Chi tiết đơn hàng
      * Mã đơn hàng
      * Mã Sản phẩm
      * Số lượng
      * Giá bán
    - Thanh toán:
      * Mã thanh toán
      * Ngày thanh toán
      * Số tiền thanh toán
      * Phương thức thanh toán
* Sơ đồ ERD của dự án:

A diagram of a network

Description automatically generated

1. Các kỹ thuật thiết kế hành vi phần mềm
   1. Thảo luận về thiết kế hành vi phần mềm (P7):
      * Phần mềm có chức năng gì?
        + Quản lý sản phẩm
        + Quản lý đơn hàng
        + Quản lý khách hàng
        + Quản lý thanh toán
        + Quản lý nhân viên
        + Quản lý hệ thống
      * Phần mềm được sử dụng các công cụ nào để phát triển (các phần mềm, ngôn ngữ lập trình được sử dụng, công cụ dùng để thiết kế phần mềm).
        + Ngôn ngữ lập trình PHP
          - là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát.
        + Thư viện Composer:
          - là một Dependency Management trong PHP, công cụ quản lý các thư viện mà project Php của bạn sử dụng. Một cách chính xác hơn **Composer** quản lý sự phụ thuộc các tài nguyên trong dự án. Nó cho phép khai báo các thư viện mà dự án của bạn sử dụng, **composer** sẽ tự động tải code của các thư viện. Nó tạo ra các file cần thiết vào project của bạn, và cập nhật các thư viện khi có phiên bản mới.
        + Cơ sở dữ liệu MySQL
          - là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System) mã nguồn mở. Nó sử dụng ngôn ngữ SQL (Structured Query Language) để quản lý và thao tác dữ liệu. MySQL thường được sử dụng để lưu trữ, quản lý dữ liệu trong các ứng dụng web, như các hệ thống quản lý nội dung (CMS), diễn đàn, trang thương mại điện tử, v.v.
        + Thư viện Laravel
          - là một framework PHP mã nguồn mở, được thiết kế để phát triển các ứng dụng web theo mô hình MVC (Model-View-Controller). Nó được tạo ra bởi Taylor Otwell vào năm 2011 và trở nên phổ biến nhờ sự đơn giản, mạnh mẽ, và nhiều tính năng hữu ích. Laravel giúp các nhà phát triển xây dựng ứng dụng web nhanh chóng và dễ dàng hơn bằng cách cung cấp các công cụ, thư viện và cấu trúc sẵn có.
        + Ứng dụng PHP storm
          - là một môi trường phát triển tích hợp (IDE - Integrated Development Environment) được thiết kế đặc biệt cho việc phát triển các ứng dụng bằng ngôn ngữ PHP. Được phát triển bởi JetBrains, PHPStorm hỗ trợ lập trình PHP cùng nhiều tính năng mạnh mẽ, giúp lập trình viên làm việc hiệu quả hơn.
        + Xampp
          - là một phần mềm mã nguồn mở giúp cài đặt và chạy một môi trường máy chủ web trên máy tính cục bộ. Tên XAMPP là viết tắt của các thành phần chính mà nó bao gồm:

**X**: Để biểu thị cho khả năng tương thích đa nền tảng (có thể cài đặt trên Windows, Linux, MacOS).

**A**: Apache (máy chủ web).

**M**: MySQL hoặc MariaDB (hệ quản trị cơ sở dữ liệu).

**P**: PHP (ngôn ngữ lập trình phía server).

**P**: Perl (một ngôn ngữ lập trình khác, ít phổ biến hơn PHP).

* + - * Ứng dụng Navicat (Quản lý database)
        + là một công cụ quản lý cơ sở dữ liệu mạnh mẽ, được thiết kế để làm việc với nhiều hệ quản trị cơ sở dữ liệu khác nhau như MySQL, MariaDB, SQL Server, Oracle, PostgreSQL, và SQLite. Nó cung cấp giao diện trực quan, giúp các nhà phát triển và quản trị viên cơ sở dữ liệu dễ dàng kết nối, quản lý, và tương tác với cơ sở dữ liệu
        + Thư viện Bootstrap 5

là phiên bản mới nhất của Bootstrap, một framework HTML, CSS, và JavaScript phổ biến được sử dụng để thiết kế giao diện người dùng (UI) cho các trang web và ứng dụng web. Bootstrap giúp các lập trình viên tạo ra giao diện đáp ứng (responsive), dễ tùy chỉnh mà không cần phải viết nhiều mã CSS hoặc JavaScript từ đầu.

* 1. **Phân tích công cụ, hành vi phần mềm (M5)**
* Trình bày 2 phương pháp đặc tả hành vi phần mềm, minh họa cho phương pháp đó bằng ví dụ.

**+** Phương pháp Sơ đồ Trạng thái (State Diagrams)

**+** Phương pháp Trường hợp Sử dụng (Use Case)