**TASK 3**

Các công đoạn của quy trình phát triển phần mềm:

1. **Giai đoạn 1: Phân tích dự án**

* Phân tích dự án để biết chi tiết về yêu cầu và mong muốn của khách hàng
* Sơ đồ Use Case
* Sơ đồ use case là một kỹ thuật được dùng trong kỹ thuật phần mềm và hệ thống để nắm bắt yêu cầu chức năng hệ thống. Use Case mô tả sự tương tác đặc trưng giữa người dùng bên ngoài (Actor) và hệ thống. Sơ đồ use case bao gồm **Actor**, **Use Case**, **Relationship.**
* Cách vẽ sơ đồ use case:
* Sơ đồ Use Case hoàn chỉnh cần trải qua 3 giai đoạn: **Giai đoạn mô hình hóa, giai đoạn cấu trúc & giai đoạn review**.
* **Giai đoạn mô hình hóa:**
* Bước 1: Thực hiện thiết lập ngữ cảnh của hệ thống.
* Bước 2: Xác định các Actor.
* Bước 3: Xác định các Use Case.
* Bước 4: Định nghĩa các mối quan hệ giữa Actor và Use Case.
* Bước 5: Đánh giá các mối quan hệ đó để tìm cách chi tiết hóa.
* **Giai đoạn cấu trúc:**
* Bước 6: Đánh giá các Use Case cho quan hệ Include.
* Bước 7: Đánh giá các Use Case cho quan hệ Extend.
* Bước 8: Đánh giá các Use Case cho quan hệ Generalization .
* **Giai đoạn review:**
* Bước 9: Kiểm tra (verification): Đảm bảo hệ thống đúng với tài liệu đặc tả.
* Bước 10: Thẩm định (validation): Đảm bảo hệ thống sẽ được phát triển là thứ mà khách hàng cuối thực sự cần thiết.
* Các tool vẽ sơ đồ Use Case: Untitled Diagram.drawio
* **Lưu ý 1: Không đặt tên Use Case quá dài.**Tên của Use Case phải đặt ngắn gọn, rõ ràng, miêu tả đủ nghĩa đối tượng người dùng.
* **Lưu ý 2: Không lạm dụng từ “Manage” quá nhiều**. Sử dụng từ “Manage” để miêu tả mối quan hệ giữa Actor và hệ thống là quá tối nghĩa, không nói rõ được mục đích để làm gì. Use Case cần truyền tải được mục đích sau cùng, chứa đựng góc nhìn của người dùng cuối cùng.
* **Lưu ý 3: Không thiết kế quá nhiều Use Case**. Tận dụng các Relationship để các Use Case liên kết với nhau. Sau đó sử dụng Boundary of System để phân nhóm, giới hạn cho các Use Case.
* **Lưu ý 4: Không quá đi sâu vào chi tiết các chức năng CRUD**. Không lạm dụng CRUD tránh gây nhàm chán, không thể hiện được nhiều thông tin cho người xem. Hướng giải quyết: Thêm một dòng note trước đoạn mô tả Use Case của tài liệu hoặc tạo riêng một Use Case có tên là manage X với X là 1 đối tượng bất kỳ.
* **Lưu ý 5: Nên tạo tính thẩm mỹ cho Use Case**. Use Case không thiết kế hợp lý, thiếu thẩm mỹ sẽ không thu hút được người dùng. Người xây dựng cần thiết kế các Use Case rõ ràng mạch lạc bằng cách: Tạo sơ đồ cùng kích cỡ, đánh dấu các Use Case ID, không chồng chéo các mối quan hệ, thêm màu sắc vào các Use Case...
* **SRS**
* SRS là từ viết tắt của Software Requirement Specification
* SRS được sử dụng để mô tả chi tiết các yêu cầu chức năng và phi chức năng của hệ thống
* Tầm quan trọng của tài liệu SRS:
* Giúp cho các bên thứ ba - stakeholders đều hiểu được hệ thống theo cùng một hướng, tránh trường hợp mỗi người một ý.
* Giúp cho đội phát triển xây dựng hệ thống một cách chính xác, đặc tả được các tính năng, không đi lạc hướng so với yêu cầu của khách hàng.
* SRS giúp nhà kiểm thử hệ thống đọc hiểu từ đó xây dựng nên kịch bản kiểm thử chi tiết nhất.
* Giúp cho việc bảo trì hệ thống và cải tiến những chức năng của hệ thống một cách nhanh chóng và dễ dàng.
* Thành phần chính của tài liệu SRS:
* Phần giới thiệu (Introduction)
* Yêu cầu mức tổng thể (High level requirement)
* Yêu cầu về bảo mật (Security Requirement)
* Đặc tả Use Case (Use Case specification)
* Thiết kế màn hình (Wireframe)
* Các yêu cầu khác (Other requirement)
* Yêu cầu tích hợp (Integration)
* Phụ lục (Appendices)
* Sơ đồ triển khai: mô hình thác nước (Waterfall model)

1. **Giai đoạn 2: Thiết kế**

* Sơ đồ kiến trúc công nghệ của ứng dụng:

## Báo cáo về giai đoạn Thiết kế dự án phần mềm

**Nội dung báo cáo (gợi ý):**

* Sơ đồ kiến trúc công nghệ của ứng dụng: là gì, tại sao phải có
* Sơ đồ ERD1, ERD2: là gì , viết tắt của gì, tại sao phải có, ba thành phần của một ERD, cách vẽ, Ví dụ.
* Chi tiết các thực thể: viết gì trong đó, cho ví dụ
* Thiếk kế giao diện: Các nguyên tắc vàng/các chú ý quan trọng, các phần mềm thiết kế giao diện miễn phí