BÀI 1 :

1. **Giao dịch bán hàng → TPS (Transaction Processing System)**

TPS là hệ thống giúp **ghi lại và xử lý các giao dịch hàng ngày**.

Khách mua hàng → hệ thống ghi lại hóa đơn, cập nhật tồn kho, lưu thông tin thanh toán.

Giống như **máy tính tiền ở siêu thị**, hoặc **phần mềm bán hàng KiotViet, Misa**.

1. Phân tích xu hướng kinh doanh → DSS (Decision Support System)

DSS giúp **phân tích dữ liệu để hỗ trợ nhà quản lý ra quyết định**.

Xem sản phẩm nào bán chạy nhất tháng này.

Phân tích dữ liệu bán hàng, doanh thu, lợi nhuận.

1. Bảng tổng quan cho CEO → EIS (Executive Information System)

EIS là hệ thống giúp **sếp cấp cao (CEO, giám đốc)** xem thông tin tổng quát để **ra quyết định chiến lược**.

Xem **báo cáo doanh thu, chi phí, lợi nhuận** qua biểu đồ.

Xem **tốc độ tăng trưởng hàng tháng** hoặc **mục tiêu quý**.

Dùng **bảng điều khiển (dashboard)** như Power BI, Tableau,…

Bai2:

## ****Dự án A: Phần mềm quản lý điểm cho trường cấp 2 – yêu cầu rõ ràng****

**Chọn mô hình:** **Waterfall (Thác nước)**

### **Lý do chọn:**

Yêu cầu **đã rõ ràng, ổn định, ít thay đổi**.

Quy trình Waterfall **tuần tự** (Phân tích → Thiết kế → Lập trình → Kiểm thử → Triển khai → Bảo trì) phù hợp cho các dự án **có kế hoạch cố định**.

Trường học là môi trường ổn định, **không cần thay đổi liên tục** trong quá trình phát triển.

**Tóm lại:** Waterfall phù hợp vì dễ quản lý, ít rủi ro, yêu cầu cụ thể ngay từ đầu.

## ****Dự án B: Ứng dụng mobile đặt lịch khám bệnh – yêu cầu linh hoạt****

**Chọn mô hình:** **Agile**

### **Lý do chọn:**

Ứng dụng mobile thường **thay đổi giao diện, tính năng, phản hồi người dùng nhanh**.

Agile cho phép **làm theo vòng lặp ngắn (Sprint)** → có thể **test, nhận phản hồi và chỉnh sửa liên tục**.

Phù hợp khi khách hàng hoặc thị trường **thường xuyên thay đổi yêu cầu**.

**Tóm lại:** Agile giúp nhóm phản ứng nhanh, ra MVP sớm, cải tiến liên tục dựa trên ý kiến người dùng.

## ****Dự án C: Hệ thống ngân hàng điện tử – bảo mật và độ phức tạp cao****

**Chọn mô hình:** **Spiral (Xoắn ốc)**

### **Lý do chọn:**

Dự án **rủi ro cao, đòi hỏi bảo mật, an toàn và độ tin cậy tuyệt đối**.

Spiral tập trung vào **phân tích và kiểm soát rủi ro ở mỗi vòng phát triển**.

Cho phép **xây dựng dần từng phần, kiểm thử và đánh giá rủi ro kỹ** trước khi mở rộng.

**Tóm lại:** Spiral giúp kiểm soát rủi ro, phù hợp cho hệ thống tài chính – ngân hàng nơi yêu cầu bảo mật và tính chính xác rất cao.

**Bai 3:**

## 5 thành phần của hệ thống đặt món ăn online

| **Thành phần** | **Mô tả** |
| --- | --- |
| **1. Con người (People)** | Bao gồm **khách hàng**, **tài xế giao hàng**, **chủ quán ăn** và **nhân viên quản trị hệ thống**.  **-Khách hàng:** đặt món, thanh toán, theo dõi đơn, đánh giá chất lượng dịch vụ.  - **Tài xế:** nhận đơn, lấy món, giao hàng, cập nhật trạng thái đơn, báo cáo sự cố.  - **Chủ quán:** quản lý thực đơn, xác nhận đơn, chuẩn bị món, theo dõi doanh thu.  - **Quản trị viên:** giám sát hệ thống, xử lý lỗi, bảo trì dữ liệu và hỗ trợ người dùng. |
| **2. Dữ liệu (Data)** | Là toàn bộ **thông tin mà hệ thống thu thập và lưu trữ** như: thông tin khách hàng, danh sách quán ăn, thực đơn, giá món, địa chỉ giao hàng, trạng thái đơn hàng, đánh giá của khách,… Dữ liệu giúp hệ thống hoạt động chính xác và đưa ra đề xuất phù hợp (ví dụ: gợi ý quán gần nhất). |
| **3. Quy trình (Procedures)** | Là **cách thức hoạt động của hệ thống**, bao gồm các **bước xử lý và luồng công việc** như: khách chọn món → đặt hàng → quán nhận đơn → tài xế đến lấy món → giao hàng → khách thanh toán. Ngoài ra còn có quy trình xử lý khiếu nại, hoàn tiền, và khuyến mãi. |
| **4. Phần mềm (Software)** | Là các **ứng dụng và chương trình** giúp con người thao tác với hệ thống.  Ví dụ: ứng dụng GrabFood trên điện thoại, phần mềm quản lý đơn hàng cho nhà hàng, hệ thống định vị GPS cho tài xế, và phần mềm máy chủ xử lý đơn hàng. |
| **5. Phần cứng (Hardware)** | Là **thiết bị vật lý** hỗ trợ hệ thống hoạt động: điện thoại của khách hàng và tài xế, máy tính của nhà hàng, máy chủ (server) lưu trữ dữ liệu, thiết bị mạng (router, modem), hệ thống GPS định vị và máy in hóa đơn tại quán. |

Một **hệ thống thông tin = Con người + Dữ liệu + Quy trình + Phần mềm + Phần cứng**

**Con người:** Ai sử dụng hệ thống

**Dữ liệu:** Hệ thống lưu gì

**Quy trình:** Hệ thống làm việc ra sao

**Phần mềm:** Công cụ xử lý

**Phần cứng:** Thiết bị hỗ trợ hoạt động

**Bài 4**

Dưới đây là bài làm hoàn chỉnh, ngắn gọn và dễ hiểu cho tình huống **“Ứng dụng điểm danh sinh viên”** theo **6 giai đoạn của SDLC (Software Development Life Cycle)**:

| **Giai đoạn** | **Việc cần làm trong dự án “Ứng dụng điểm danh sinh viên”** |
| --- | --- |
| 1. **Planning (Lập kế hoạch)** | Ở giai đoạn đầu, nhóm phát triển sẽ **xác định mục tiêu dự án**, ví dụ: xây dựng ứng dụng giúp giảng viên điểm danh nhanh và lưu kết quả chính xác. Tiếp theo là **xác định phạm vi**, nguồn lực, thời gian, ngân sách và **phân chia công việc** cho từng thành viên trong nhóm. Mục tiêu của giai đoạn này là tạo ra **kế hoạch tổng thể** cho toàn dự án. |
| **2. Analysis (Phân tích yêu cầu)** | Nhóm sẽ **thu thập thông tin** từ người dùng thật: giảng viên, sinh viên, và phòng đào tạo. Từ đó, xác định **các chức năng cần có** như: đăng nhập, xem danh sách lớp, điểm danh, thống kê số buổi học, xuất báo cáo. Giai đoạn này cũng phân tích **yêu cầu dữ liệu** (bảng sinh viên, môn học, lớp học) và **yêu cầu kỹ thuật** để đảm bảo ứng dụng hoạt động đúng với nhu cầu thực tế. |
| **3. Design (Thiết kế)** | Sau khi có yêu cầu rõ ràng, nhóm sẽ **thiết kế giao diện người dùng (UI)** cho cả giảng viên và sinh viên, đảm bảo dễ sử dụng. Tiếp đó, thiết kế **cấu trúc cơ sở dữ liệu** gồm các bảng: Sinh viên, Lớp học, Môn học, Lịch học, Trạng thái điểm danh. Ngoài ra, nhóm tạo **sơ đồ Use Case**, **biểu đồ luồng dữ liệu (DFD)**, và **prototype (mẫu giao diện)** để trình bày cách hệ thống hoạt động. |
| **4. Implementation (Lập trình / Cài đặt)** | Ở bước này, lập trình viên **bắt đầu viết mã nguồn** cho từng chức năng: đăng nhập, điểm danh, lưu dữ liệu, hiển thị danh sách lớp, thống kê buổi học. Các module được xây dựng dần và **kết nối với cơ sở dữ liệu**. Ví dụ: giảng viên đăng nhập → quét mã QR sinh viên → dữ liệu điểm danh được lưu vào hệ thống. Sau khi lập trình xong, nhóm sẽ tích hợp các phần lại thành một hệ thống hoàn chỉnh. |
| **5. Testing (Kiểm thử)** | Khi ứng dụng đã hoàn thành, nhóm tiến hành **kiểm thử** để đảm bảo hoạt động đúng như yêu cầu. Gồm các bước:  Kiểm thử đơn vị (Unit Test) cho từng chức năng nhỏ.  Kiểm thử tích hợp (Integration Test) giữa các phần.  Kiểm thử với người dùng thật (User Acceptance Test) để lấy phản hồi. Nếu phát hiện lỗi, nhóm sẽ **sửa và tối ưu** hiệu năng trước khi triển khai. |
| **6. Deployment & Maintenance (Triển khai và Bảo trì)** | Cuối cùng, ứng dụng được **cài đặt và triển khai chính thức** cho giảng viên và sinh viên trong trường. Nhóm sẽ **hướng dẫn sử dụng**, theo dõi hoạt động, tiếp nhận phản hồi. Sau khi triển khai, giai đoạn **bảo trì** sẽ diễn ra thường xuyên để **cập nhật, sửa lỗi, thêm tính năng mới** (như đồng bộ với hệ thống đào tạo, thêm thông báo tự động,…). |

### Tóm tắt dễ nhớ:

SDLC gồm 6 giai đoạn chính:  
**Planning → Analysis → Design → Implementation → Testing → Deployment & Maintenance**

**BÀI 5 :**

Dưới đây là phần trình bày chi tiết, đúng theo yêu cầu bài **“Áp dụng SDLC cho hệ thống điểm danh sinh viên bằng QR code”**, tập trung vào **3 giai đoạn đầu của SDLC**:

## ****1. Planning (Lập kế hoạch)****

**Mục tiêu:**  
Xác định phạm vi, mục đích, nguồn lực và kế hoạch thực hiện dự án.

**Công việc thực hiện:**

**Xác định mục tiêu:** Xây dựng hệ thống giúp giảng viên điểm danh sinh viên nhanh chóng bằng mã QR, hạn chế gian lận và tiết kiệm thời gian.

**Phạm vi dự án:**

Giảng viên tạo buổi học, hệ thống sinh mã QR.

Sinh viên quét mã QR để điểm danh.

Phòng đào tạo xem báo cáo thống kê.

**Lập kế hoạch nguồn lực:** Phân chia công việc cho nhóm:

1 người phân tích yêu cầu,

2 người lập trình backend/frontend

1 người thiết kế giao diện,

1 người kiểm thử.

**Thời gian dự kiến:** 8–10 tuần (2 tuần/pha phát triển).

**Công cụ và công nghệ:** React Native hoặc Flutter cho ứng dụng sinh viên, Node.js hoặc Java Spring Boot cho backend, cơ sở dữ liệu MySQL hoặc Firebase.

## ****2. Requirement Analysis (Phân tích yêu cầu)****

**Mục tiêu:**  
Thu thập, xác định và làm rõ các yêu cầu của người dùng và hệ thống.

**Công việc thực hiện:**

**Thu thập yêu cầu:**

**Giảng viên:** cần tạo buổi học, sinh mã QR, xem danh sách sinh viên có mặt/vắng.

**Sinh viên:** đăng nhập bằng tài khoản, quét mã QR để điểm danh.

**Phòng đào tạo:** xem thống kê điểm danh theo lớp, học phần, thời gian.

**Phân loại yêu cầu:**

**Yêu cầu chức năng:** tạo buổi học, sinh mã QR, quét mã, lưu dữ liệu điểm danh, báo cáo thống kê.

**Yêu cầu phi chức năng:** bảo mật (xác thực tài khoản), tốc độ xử lý nhanh, tương thích trên nhiều thiết bị.

**Xây dựng sơ đồ UML tương ứng:**

**Use Case Diagram:** mô tả các tác nhân (Giảng viên, Sinh viên, Phòng đào tạo) và các chức năng mà họ sử dụng.

**Activity Diagram:** mô tả luồng quy trình “Giảng viên tạo buổi học → Sinh viên quét mã → Hệ thống ghi nhận → Phòng đào tạo xem thống kê”.

## ****3. System Design (Thiết kế hệ thống)****

**Mục tiêu:**  
Tạo bản thiết kế chi tiết cho hệ thống dựa trên các yêu cầu đã phân tích.

**Công việc thực hiện:**

**Thiết kế kiến trúc hệ thống:**

Ứng dụng di động cho sinh viên (quét mã QR).

Ứng dụng web cho giảng viên và phòng đào tạo (tạo buổi học, xem thống kê).

Máy chủ backend xử lý logic, lưu dữ liệu và xác thực người dùng.

**Thiết kế cơ sở dữ liệu:**

Bảng SinhVien (Mã SV, Họ tên, Lớp).

Bảng GiangVien (Mã GV, Họ tên, Bộ môn).

Bảng LopHoc (Mã lớp, Tên môn học, Lịch học).

Bảng DiemDanh (Mã buổi học, Mã SV, Thời gian quét, Trạng thái).

**Thiết kế giao diện (UI):**

Màn hình giảng viên: tạo buổi học, hiển thị mã QR, xem danh sách sinh viên.

Màn hình sinh viên: đăng nhập, quét mã QR, xem lịch sử điểm danh.

Màn hình phòng đào tạo: bảng thống kê tổng hợp.

**Thiết kế sơ đồ UML:**

**Class Diagram:** mô tả các lớp SinhVien, GiangVien, BuoiHoc, DiemDanh.

**Sequence Diagram:** mô tả chuỗi tương tác “Giảng viên → Hệ thống → Sinh viên → CSDL”.

**Bài 6:**

| **Tình huống** | **Sơ đồ UML phù hợp** | **Giải thích** |
| --- | --- | --- |
| **A. Mô tả chức năng người dùng có thể thực hiện trên ứng dụng học tiếng Anh** | **Use Case Diagram (Sơ đồ ca sử dụng)** | Dùng để **mô tả các chức năng (use case)** mà hệ thống cung cấp cho **người dùng (actors)**. Ví dụ: học viên có thể “Đăng nhập”, “Học từ vựng”, “Làm bài kiểm tra”, “Xem kết quả”. |
| **B. Mô tả lớp NguoiDung, KhoaHoc, BaiHoc và quan hệ giữa chúng** | **Class Diagram (Sơ đồ lớp)** | Dùng để **mô tả cấu trúc tĩnh của hệ thống**: các **lớp (class)**, **thuộc tính (attributes)**, **phương thức (methods)** và **mối quan hệ** giữa chúng (kế thừa, kết hợp, liên kết). |
| **C. Mô tả luồng học viên bắt đầu → vào học → làm bài → hoàn thành** | **Activity Diagram (Sơ đồ hoạt động)** | Dùng để **mô tả luồng xử lý công việc (workflow)** hoặc **quy trình nghiệp vụ** theo thứ tự các bước. Ở đây là quy trình học của học viên. |
| **D. Mô tả cách hệ thống triển khai trên các máy chủ, thiết bị** | **Deployment Diagram (Sơ đồ triển khai)** | Dùng để **mô tả cách các thành phần phần mềm được triển khai lên phần cứng** — như máy chủ, thiết bị di động, mạng, cơ sở dữ liệu. |
| **E. Mô tả thứ tự tương tác giữa học viên và hệ thống khi nộp bài** | **Sequence Diagram (Sơ đồ tuần tự)** | Dùng để **mô tả thứ tự các thông điệp, tương tác giữa các đối tượng hoặc thành phần** trong quá trình thực hiện một chức năng cụ thể (ví dụ: nộp bài → gửi dữ liệu → xác nhận → hiển thị kết quả). |

Baif7 :

6 giai đoạn của **vòng đời phát triển phần mềm (SDLC)** áp dụng cho **hệ thống đăng ký tiêm chủng online** tại trung tâm y tế:

## ****1. Planning (Lập kế hoạch)****

**Mục tiêu:**  
Xác định phạm vi, mục tiêu, thời gian và nguồn lực của dự án.

**Công việc thực hiện:**

Xác định **mục tiêu chính**: Xây dựng hệ thống giúp người dân **đăng ký tiêm chủng online**, **trung tâm y tế xác nhận và quản lý lịch tiêm**.

**Phạm vi dự án:**

Người dân: đăng ký thông tin, chọn loại vắc xin, đặt lịch tiêm.

Trung tâm y tế: xác nhận đăng ký, sắp lịch tiêm, quản lý người đã tiêm/chưa tiêm.

**Lập kế hoạch nhân sự:**

1 trưởng nhóm quản lý dự án,

2 lập trình viên,

1 chuyên viên thiết kế UI/UX,

1 kiểm thử viên (tester).

**Xác định công nghệ sử dụng:**

Website hoặc ứng dụng mobile,

Cơ sở dữ liệu MySQL hoặc Firebase,

Backend dùng Node.js hoặc Java Spring Boot.

## ****2. Requirement Analysis (Phân tích yêu cầu)****

**Mục tiêu:**  
Thu thập và xác định các yêu cầu của người dùng và hệ thống.

**Công việc thực hiện:**

**Thu thập yêu cầu:**

**Người dân:** đăng ký tiêm, nhập thông tin cá nhân, chọn mũi tiêm, nhận thông báo lịch hẹn.

**Trung tâm:** xác nhận đăng ký, sắp lịch tiêm, cập nhật tình trạng tiêm, thống kê.

**Xác định yêu cầu chức năng:**

Đăng ký tài khoản, đăng ký tiêm, xác nhận lịch, tra cứu trạng thái, thống kê theo khu vực.

**Xác định yêu cầu phi chức năng:**

Hệ thống phải bảo mật, hoạt động 24/7, dễ sử dụng và phản hồi nhanh.

**Công cụ UML:**

**Use Case Diagram** để mô tả chức năng giữa Người dân – Trung tâm – Hệ thống.

**Activity Diagram** mô tả quy trình đăng ký → xác nhận → tiêm → thống kê.

## ****3. System Design (Thiết kế hệ thống)****

**Mục tiêu:**  
Thiết kế giao diện, cấu trúc dữ liệu và mô hình hệ thống tổng thể.

**Công việc thực hiện:**

**Thiết kế kiến trúc:**

Website có hai giao diện:

Người dân: đăng ký, xem lịch, nhận thông báo.

Trung tâm: xác nhận, thống kê, báo cáo.

**Thiết kế cơ sở dữ liệu:**

Bảng NguoiDan (Họ tên, Ngày sinh, CCCD, SĐT, Email).

Bảng DangKyTiem (Mã đăng ký, Mã người dân, Loại vắc xin, Ngày mong muốn, Trạng thái).

Bảng LichTiem (Ngày tiêm, Cơ sở tiêm, Bác sĩ phụ trách).

**Thiết kế giao diện (UI):**

Màn hình đăng ký, màn hình xác nhận, màn hình theo dõi danh sách đã tiêm.

**Sơ đồ UML:**

**Class Diagram**: mô tả các lớp như NguoiDan, DangKyTiem, LichTiem.

**Sequence Diagram**: mô tả quá trình người dân đăng ký và trung tâm xác nhận lịch.

## ****4. Implementation (Lập trình / Cài đặt)****

**Mục tiêu:**  
Hiện thực hóa thiết kế thành sản phẩm phần mềm thực tế.

**Công việc thực hiện:**

Xây dựng **giao diện web/app** cho người dân và trung tâm y tế.

**Lập trình backend** để xử lý đăng ký, xác nhận, gửi thông báo tự động qua email/SMS.

Kết nối **cơ sở dữ liệu** để lưu trữ thông tin người đăng ký, lịch tiêm, và trạng thái tiêm.

Hoàn thiện chức năng **tra cứu, thống kê** cho quản lý trung tâm.

## ****5. Testing (Kiểm thử)****

**Mục tiêu:**  
Đảm bảo hệ thống hoạt động đúng yêu cầu, không lỗi và ổn định.

**Công việc thực hiện:**

**Kiểm thử đơn vị (Unit Test):** kiểm tra từng chức năng nhỏ như đăng nhập, đăng ký, xác nhận lịch.

**Kiểm thử tích hợp (Integration Test):** kiểm tra luồng hoạt động giữa các chức năng liên quan.

**Kiểm thử giao diện (UI Test):** đảm bảo người dùng thao tác dễ dàng.

**Kiểm thử bảo mật:** xác minh đăng nhập, phân quyền truy cập.

**Kiểm thử người dùng thực tế (User Acceptance Test):** cho nhân viên y tế và người dân thử nghiệm.

## ****6. Deployment & Maintenance (Triển khai và Bảo trì)****

**Mục tiêu:**  
Đưa hệ thống vào sử dụng thực tế và duy trì ổn định lâu dài.

**Công việc thực hiện**

**Triển khai:**

Cài đặt hệ thống lên máy chủ (server) và chạy thử nghiệm tại trung tâm y tế.

Hướng dẫn nhân viên sử dụng.

**Giám sát hoạt động:**

Theo dõi hiệu suất, phản hồi người dùng.

Xử lý lỗi, vá bảo mật.

**Bảo trì, nâng cấp:**

Cập nhật thêm tính năng như gửi tin nhắn nhắc lịch, tích hợp bản đồ vị trí tiêm.

Mở rộng hệ thống cho nhiều trung tâm y tế khác nếu cần.

Bai 8:

## ****1. Xác định các tác nhân chính và chức năng tương ứng****

| **Tác nhân (Actor)** | **Chức năng chính trong hệ thống** |
| --- | --- |
| **Học viên (Student)** | - Đăng ký tài khoản, đăng nhập. - Xem và đăng ký khóa học. - Tham gia bài học, làm bài kiểm tra, nộp bài. - Xem điểm và phản hồi từ giảng viên. |
| **Giảng viên (Teacher)** | - Tạo và quản lý bài học, tài liệu giảng dạy. - Chấm điểm, ghi nhận kết quả học tập. - Theo dõi tiến độ của học viên trong lớp. - Gửi thông báo hoặc phản hồi học tập. |
| **Quản trị viên (Admin)** | - Quản lý người dùng (thêm/xóa học viên, giảng viên). - Phân quyền sử dụng hệ thống. - Theo dõi báo cáo, thống kê khóa học và kết quả. - Quản lý nội dung, khóa học, tài khoản và bảo mật hệ thống. |

## ****2. Phân loại loại hệ thống thông tin phù hợp****

Nền tảng học trực tuyến này **tích hợp nhiều loại hệ thống thông tin**, cụ thể:

| **Loại hệ thống thông tin** | **Ứng dụng trong hệ thống học trực tuyến** |
| --- | --- |
| **TPS (Transaction Processing System)** | Xử lý các giao dịch hằng ngày như: học viên đăng ký khóa học, nộp bài, lưu điểm, cập nhật hồ sơ. |
| **MIS (Management Information System)** | Hỗ trợ giảng viên và admin tạo báo cáo, thống kê kết quả học tập, điểm trung bình, tỷ lệ hoàn thành khóa học. |
| **DSS (Decision Support System)** | Phân tích dữ liệu để hỗ trợ nhà quản lý ra quyết định, ví dụ: xác định lớp học hiệu quả, giảng viên được đánh giá cao, đề xuất cải tiến nội dung. |

→ **Tóm lại:** hệ thống học trực tuyến này là **sự kết hợp giữa TPS, MIS và DSS**, trong đó **TPS** là nền tảng cốt lõi vì xử lý giao dịch học tập hằng ngày.

## ****3. Đề xuất mô hình phát triển phần mềm phù hợp nhất****

**Mô hình đề xuất:** **Agile**

**Lý do chọn Agile:**

Yêu cầu của người dùng (giảng viên, học viên, quản trị) **thay đổi thường xuyên**, cần linh hoạt để cập nhật giao diện, chức năng học, và tính năng báo cáo.

**Phát triển theo vòng lặp (Sprint)** giúp ra mắt sớm phiên bản thử nghiệm (MVP) → nhận phản hồi sớm từ người dùng thật.

Thích hợp cho dự án web/app hiện đại có đội ngũ nhỏ, giao tiếp liên tục.

Cho phép **phát triển song song nhiều tính năng**: nhóm 1 làm phần học viên, nhóm 2 làm phần giảng viên, nhóm 3 làm admin dashboard.

**Ví dụ vòng lặp Agile:**

Sprint 1: Đăng nhập, đăng ký khóa học.

Sprint 2: Học và nộp bài.

Sprint 3: Quản lý điểm và thống kê.

Sprint 4: Báo cáo và phân quyền.

## ****4. Ba sơ đồ UML sẽ sử dụng trong thiết kế hệ thống****

| **Sơ đồ UML** | **Mục đích sử dụng** |
| --- | --- |
| **Use Case Diagram (Sơ đồ ca sử dụng)** | Mô tả các chức năng mà học viên, giảng viên và admin có thể thực hiện trên hệ thống. |
| **Class Diagram (Sơ đồ lớp)** | Mô tả cấu trúc dữ liệu, các lớp như HocVien, KhoaHoc, BaiHoc, GiangVien, BangDiem và mối quan hệ giữa chúng. |
| **Activity Diagram (Sơ đồ hoạt động)** | Mô tả luồng hoạt động như “Học viên đăng ký khóa học → Tham gia → Làm bài → Nhận điểm”. |

Bai9:

**phân tích toàn diện hệ thống quản lý đơn hàng cho doanh nghiệp logistic**, kết hợp đầy đủ kiến thức về **SDLC – hệ thống thông tin – mô hình phát triển – UML**.

## ****1. Xác định các tác nhân (Actors) và chức năng chính****

| **Tác nhân (Actor)** | **Chức năng chính** |
| --- | --- |
| **Khách hàng** | - Tạo đơn hàng mới (địa chỉ, hàng hóa, người nhận). - Theo dõi trạng thái đơn hàng (đang giao, đã giao, bị hủy). - Nhận thông báo khi đơn hàng thay đổi trạng thái. |
| **Nhân viên vận chuyển (Shipper)** | - Nhận đơn hàng cần giao. - Cập nhật trạng thái: đã nhận hàng, đang giao, giao thành công/thất bại. - Báo cáo sự cố (không liên lạc được khách, hàng lỗi...). |
| **Quản lý trung tâm (Manager)** | - Theo dõi danh sách đơn hàng trong khu vực. - Xem hiệu suất giao hàng theo ngày, tuần, tháng. - Phát hiện khu vực có tồn đọng hoặc tỉ lệ trễ cao. |
| **CEO / Ban lãnh đạo** | - Xem tổng quan toàn hệ thống: doanh thu, số đơn, tỉ lệ giao đúng hạn, bản đồ hoạt động. - Phân tích dữ liệu để ra quyết định chiến lược (mở rộng vùng, tối ưu nhân lực). |

## ****2. Phân loại từng chức năng vào hệ thống thông tin phù hợp****

| **Chức năng** | **Loại hệ thống thông tin (TPS / MIS / DSS / EIS)** | **Giải thích** |
| --- | --- | --- |
| Khách hàng tạo đơn hàng, theo dõi đơn | **TPS (Transaction Processing System)** | Xử lý giao dịch hàng ngày: tạo đơn, cập nhật, theo dõi trạng thái. |
| Nhân viên cập nhật trạng thái giao hàng | **TPS** | Ghi nhận và cập nhật các thay đổi trực tiếp trong hệ thống theo thời gian thực. |
| Quản lý xem báo cáo, khu vực tồn đọng | **MIS (Management Information System)** | Hệ thống tổng hợp dữ liệu từ TPS, cung cấp báo cáo, thống kê để quản lý ra quyết định vận hành. |
| CEO xem tổng quan, biểu đồ, vùng hoạt động | **EIS (Executive Information System)** | Hệ thống dành cho lãnh đạo cấp cao, hiển thị thông tin tổng hợp (dashboard, KPI, biểu đồ) giúp đưa ra quyết định chiến lược. |
| Phân tích hiệu suất giao hàng, dự đoán khu vực trễ | **DSS (Decision Support System)** | Phân tích dữ liệu từ các báo cáo để hỗ trợ ra quyết định nâng cao (tối ưu tuyến đường, phân bổ nhân lực). |

## ****3. Chọn mô hình phát triển phần mềm phù hợp và giải thích****

**Mô hình đề xuất:** **Agile – Phát triển linh hoạt**

**Lý do chọn Agile:**

Dự án logistic có **nhiều loại người dùng** (khách hàng, shipper, quản lý, CEO) → yêu cầu thay đổi thường xuyên.

**Agile** cho phép phát triển theo **vòng lặp ngắn (Sprint)**, có thể **thử nghiệm từng phần** như: tạo đơn, theo dõi đơn, dashboard quản lý,...

Giúp **ra mắt sớm sản phẩm khả dụng (MVP)** → nhận phản hồi từ thực tế để điều chỉnh.

Dễ **mở rộng tính năng**: ví dụ thêm chatbot thông báo, bản đồ giao hàng trực tuyến, thống kê nâng cao

**Ví dụ triển khai theo Sprint:**

| **Sprint** | **Tính năng chính** |
| --- | --- |
| Sprint 1 | Tạo đơn và theo dõi trạng thái (TPS) |
| Sprint 2 | Quản lý báo cáo, thống kê (MIS) |
| Sprint 3 | Dashboard tổng quan (EIS) |
| Sprint 4 | Phân tích dữ liệu nâng cao (DSS) |

## ****4. Đề xuất 4 sơ đồ UML sử dụng trong phân tích và thiết kế****

| **Tên sơ đồ UML** | **Mục đích sử dụng** |
| --- | --- |
| **Use Case Diagram (Sơ đồ ca sử dụng)** | Mô tả các chức năng chính và mối quan hệ giữa các tác nhân (Khách hàng, Shipper, Quản lý, CEO) và hệ thống. |
| **Activity Diagram (Sơ đồ hoạt động)** | Mô tả luồng nghiệp vụ: khách hàng tạo đơn → nhân viên giao → cập nhật trạng thái → quản lý xem báo cáo. |
| **Class Diagram (Sơ đồ lớp)** | Mô tả các lớp chính như DonHang, KhachHang, NhanVien, TrangThai, KhuVuc, cùng mối quan hệ dữ liệu giữa chúng. |
| **Sequence Diagram (Sơ đồ tuần tự)** | Mô tả trình tự tương tác khi thực hiện chức năng cụ thể, ví dụ: “Khách hàng theo dõi đơn hàng” hay “Shipper cập nhật trạng thái giao”. |

BTTH:

**Phân tích và đề xuất thiết kế hệ thống phần mềm quản lý giao hàng** cho công ty logistics — trình bày theo **chuẩn logic, rõ ràng, đầy đủ 4 yêu cầu** của bài tập.

## ****1. Xác định các tác nhân (Actors) và chức năng chính****

| **Tác nhân (Actor)** | **Chức năng chính trong hệ thống** |
| --- | --- |
| **Khách hàng (Customer)** | - Tạo đơn hàng online (nhập thông tin sản phẩm, địa chỉ nhận, hình thức thanh toán). - Theo dõi trạng thái đơn hàng theo thời gian thực (đang lấy hàng, đang giao, đã giao). - Nhận thông báo tự động khi đơn thay đổi trạng thái. |
| **Nhân viên giao hàng (Shipper)** | - Nhận danh sách đơn hàng được phân công. - Cập nhật trạng thái giao hàng (đã nhận, đang giao, giao thành công, giao thất bại). - Báo cáo sự cố trong quá trình giao hàng (không liên lạc được khách, sai địa chỉ,...). |
| **Quản lý khu vực (Manager)** | - Theo dõi danh sách đơn hàng trong khu vực phụ trách. - Xem báo cáo hiệu suất giao hàng (tỉ lệ giao đúng hạn, khu vực trễ, đơn tồn). - Phân công lại nhân viên giao hàng nếu cần. |
| **CEO / Ban lãnh đạo** | - Xem báo cáo tổng quan toàn hệ thống: doanh thu, số lượng đơn theo khu vực, tỉ lệ giao đúng hạn, hiệu suất tổng thể. - Đưa ra quyết định chiến lược: mở thêm kho, tối ưu tuyến đường, tăng nhân sự vùng chậm. |

## ****2. Phân loại từng chức năng vào các loại hệ thống thông tin phù hợp****

| **Chức năng / Nhiệm vụ** | **Loại hệ thống thông tin (TPS / MIS / DSS / EIS)** | **Giải thích** |
| --- | --- | --- |
| Tạo đơn hàng, theo dõi trạng thái | **TPS (Transaction Processing System)** | Hệ thống xử lý các giao dịch thường xuyên hàng ngày: tạo, cập nhật, lưu đơn hàng. |
| Nhân viên cập nhật tiến độ giao hàng | **TPS** | Xử lý và ghi nhận trạng thái đơn hàng trong thời gian thực. |
| Quản lý xem báo cáo khu vực, hiệu suất giao hàng | **MIS (Management Information System)** | Cung cấp báo cáo, thống kê giúp quản lý kiểm soát hoạt động vận hành. |
| CEO xem tổng quan hoạt động, tỷ lệ giao đúng hạn | **EIS (Executive Information System)** | Hệ thống hỗ trợ nhà lãnh đạo xem thông tin tổng hợp qua dashboard, biểu đồ, KPI. |
| Phân tích hiệu suất, đề xuất cải tiến tuyến đường | **DSS (Decision Support System)** | Hệ thống hỗ trợ ra quyết định bằng cách phân tích dữ liệu lịch sử, xác định vùng chậm, tối ưu nguồn lực. |

## ****3. Mô hình phát triển phần mềm phù hợp và lý do lựa chọn****

**Mô hình được đề xuất:** **Agile (Phát triển linh hoạt)**

**Lý do chọn:**

Hệ thống có **nhiều nhóm người dùng khác nhau (khách hàng, shipper, quản lý, CEO)** → yêu cầu thường xuyên thay đổi.

**Agile cho phép phát triển theo vòng lặp (Sprint)** → có thể hoàn thiện dần từng phần và nhận phản hồi liên tục từ người dùng.

Dễ dàng mở rộng, tích hợp tính năng mới như **định vị GPS, thông báo đẩy, AI gợi ý tuyến giao hàng**.

Phù hợp với dự án có quy mô vừa – lớn, cần **triển khai sớm bản thử nghiệm (MVP)** để kiểm chứng thực tế.

**Ví dụ kế hoạch theo Sprint:**

| **Sprint** | **Tính năng chính** |
| --- | --- |
| Sprint 1 | Tạo đơn hàng, theo dõi trạng thái (TPS) |
| Sprint 2 | Cập nhật tiến độ giao hàng, thông báo tự động |
| Sprint 3 | Báo cáo hiệu suất khu vực (MIS) |
| Sprint 4 | Dashboard tổng quan, phân tích dữ liệu (EIS & DSS) |

## ****4. Đề xuất 4 sơ đồ UML phù hợp và mục đích sử dụng****

| **Tên sơ đồ UML** | **Mục đích sử dụng** |
| --- | --- |
| **Use Case Diagram (Sơ đồ ca sử dụng)** | Mô tả các chức năng mà từng tác nhân (khách hàng, nhân viên, quản lý, CEO) có thể thực hiện. Giúp xác định phạm vi và yêu cầu tổng thể của hệ thống. |
| **Activity Diagram (Sơ đồ hoạt động)** | Thể hiện luồng công việc của quá trình “Tạo đơn → Giao hàng → Cập nhật trạng thái → Hoàn tất → Báo cáo”. Dùng để hiểu rõ quy trình vận hành. |
| **Class Diagram (Sơ đồ lớp)** | Mô tả cấu trúc dữ liệu chính và mối quan hệ giữa các lớp như DonHang, KhachHang, NhanVien, TrangThai, KhuVuc, BaoCao. |
| **Sequence Diagram (Sơ đồ tuần tự)** | Diễn tả trình tự tương tác giữa các đối tượng trong một chức năng cụ thể, ví dụ “Khách hàng theo dõi đơn hàng” hoặc “Nhân viên cập nhật trạng thái”. |