**Bài 1 :**

- Giao dịch bán hàng : TPS

- Phân tích xu hướng kinh doanh : DSS

- Bảng tổng quan hiệu suất hàng tháng dành cho CEO : EIS

**Bài 2 :**

- Phần mềm quản lí điểm cho trường cấp 2 ( A ) : Waterfall

=> Bởi vì nó cần yêu cầu rõ ràng.

- Ứng dụng mobile đặt lịch khám bệnh ( B ) : Agile

=> Bởi vì nó yêu cầu tính linh hoạt.

- Hệ thống ngân hàng điện tử ( C ) : Spiral

=> Bởi vì nó yêu cầu tính bảo mật và độ phức tạp cao.

**Bài 3 :**

- Con người :

+ Tài xế giao hàng

+ Nhà hàng / Quán ăn

+ Khách hàng

+ Nhân viên quản trị hệ thống

- Dữ liệu :

+ Thông tin khách hàng

+ Dữ liệu món ăn

+ Đơn hàng

+ Dữ liệu nhà hàng , tài xế , phản hồi và đánh giá

- Quy trình :

+ Khách hàng chọn món => Thêm vào giỏ hàng => Thanh toán => Nhà hàng xác nhận đơn => Tài xế nhận đơn => Giao món => Khách hàng nhận đơn và đánh giá .

- Phần mềm :

+ Ứng dụng di động ( ShoppeFood , … )

+ Trang web đặt món

+ Ứng dụng riêng cho tài xế và nhà hàng

+ Phần mềm quản trị hệ thống

- Phần cứng :

+ Máy chủ ( server ) lưu trữ dữ liệu và chạy ứng dụng

+ Điện thoại di động , máy tính bảng của khách hàng , tài xế và nhà hàng

+ Thiết bị mạng , bộ định tuyến ( router ) , máy in hóa đơn của quán ăn

**Bài 4 :**

- Planning :

+ Xác định mục tiêu của ứng dụng

+ Xác định phạm vi dự án , nguồn lực , thời gian , chi phí

+ Phân công vai trò trong nhóm

- Analysis :

+ Thu thập và phân tích yêu cầu người dùng :

+ Giáo viên muốn điểm danh qua app

+ Sinh viên muốn xem lịch sử điểm danh

+ Quản trị viên cần quản lí lớp , môn , người dùng

=> Xác định các chức năng chính : đăng nhập , tạo lớp , …

- Design :

+ Thiết kế giao diện và kiến trúc hệ thống

+ Vẽ sơ đồ use-case , ERD , wireframe giao diện

+ Thiết kế cơ sở dữ liệu

+ Xác định công nghệ

- Implementation :

+ Xây dựng chức năng đăng nhập , quản lí lớp học , điểm danh mã QR hoặc thủ công

+ Kết nói frontend-backend-database

+ Viết API để truy xuất dữ liệu

- Testing :

+ Kiểm thử đơn vị , tích hợp và giao diện

+ Phát hiện và sửa lỗi

+ Mời giáo viên thử nghiệm thực tế để lấy phản hồi

- Deployment & Maintenance :

+ Triển khai ứng dụng lên server

+ Hướng dẫn sử dụng

+ Bảo trì , cập nhật

**Bài 5 :**

1. Planning (Lập kế hoạch)

Mục tiêu: Xây dựng hệ thống điểm danh sinh viên bằng QR code.

Phạm vi:

Giảng viên: tạo buổi học, quét QR.

Sinh viên: quét QR điểm danh.

Phòng đào tạo: thống kê điểm danh.

Tài nguyên: Web app + Mobile app, Node.js, React, MySQL.

Timeline: 1 tuần lập kế hoạch, 1-2 tuần phân tích yêu cầu, 2 tuần thiết kế.

Rủi ro: QR code không quét được, lỗi đồng bộ dữ liệu, quyền truy cập chưa chuẩn.

2. Requirement Analysis (Phân tích yêu cầu)

Functional:

Giảng viên: tạo buổi học, quét QR.

Sinh viên: quét QR điểm danh.

Phòng đào tạo: thống kê điểm danh.

Non-functional: bảo mật, hoạt động mượt trên web và mobile, thời gian quét QR nhanh.

UML sơ lược:

Actors: Giảng viên, Sinh viên, Phòng đào tạo.

Use-case: Tạo buổi học, quét QR, điểm danh, thống kê.

3. System Design (Thiết kế hệ thống)

Kiến trúc:

Client (Web/Mobile) ↔ Server (API) ↔ Database (MySQL).

Class diagram đơn giản:

GiangVien: taoBuoiHoc(), quetQR()

SinhVien: diemDanh()

BuoiHoc: generateQR()

PhongDaoTao: xemThongKe()

Cơ sở dữ liệu: bảng GiangVien, SinhVien, BuoiHoc, DiemDanh.

Giao diện: Web: quản lý buổi học, quét QR, thống kê; Mobile: quét QR điểm danh.

**Bài 6 :**

Sơ đồ UML phù hợp :

1. Use-case diagram (Sơ đồ ca sử dụng)

B. Class diagram (Sơ đồ lớp)

C. Activity diagram (Sơ đồ hoạt động)

D. Deployment diagram (Sơ đồ triển khai)

E. Sequence diagram (Sơ đồ tuần tự)

**Bài 7 :**

1. Khảo sát và xác định yêu cầu (Planning & Analysis)

Tìm hiểu nhu cầu của trung tâm và người dân.

Xác định chức năng: đăng ký tiêm, xác nhận lịch, quản lý danh sách.

Phân tích tính khả thi về kỹ thuật, kinh tế, và pháp lý.

2. Phân tích hệ thống (System Analysis)

Xây dựng sơ đồ use case, DFD, mô tả quy trình nghiệp vụ.

Xác định yêu cầu chức năng và phi chức năng chi tiết.

3. Thiết kế hệ thống (System Design)

Thiết kế giao diện, cơ sở dữ liệu, kiến trúc hệ thống.

Xác định bảo mật, phân quyền, kết nối mạng.

4. Lập trình và triển khai (Implementation / Development)

Lập trình các module: đăng ký, xác nhận, thống kê.

Kết nối CSDL và kiểm thử nội bộ.

5. Kiểm thử và tích hợp (Testing & Integration)

Kiểm thử chức năng, giao diện, hiệu năng.

Sửa lỗi, đảm bảo hệ thống hoạt động ổn định trước khi triển khai thực tế.

6. Bảo trì và nâng cấp (Maintenance)

Theo dõi, khắc phục sự cố, cập nhật tính năng mới.

Bảo đảm hệ thống luôn an toàn và phù hợp với quy định y tế.

**Bài 8 :**

1) Tác nhân chính & chức năng tương ứng

Học viên (Student)

Đăng ký/đăng nhập, duyệt khóa học, ghi danh, xem video/bài tập, nộp bài, xem điểm, tham gia lớp trực tuyến, chat/QA.

Giảng viên (Instructor)

Tạo/chuẩn bị bài học, đăng bài giảng, giao/ chấm bài, nhập điểm, quản lý lịch dạy, tương tác với học viên.

Admin (Quản trị hệ thống / Trung tâm)

Quản lý người dùng, phân quyền, duyệt/hoàn tiền, cấu hình khóa học, theo dõi báo cáo & thống kê, quản lý tài nguyên (video, tài liệu).

Hệ thống thanh toán / Cổng thanh toán (Payment)

Xử lý thanh toán, phát hành hoá đơn, hoàn tiền.

Hạ tầng bên ngoài (Video CDN / Zoom / Email/SMS)

Streaming, hội họp trực tuyến, gửi thông báo.

2) Phân loại hệ thống thông tin (theo chức năng)

TPS (Transaction Processing System)

Xử lý giao dịch hàng ngày: đăng ký học, thanh toán, nộp bài, điểm danh buổi học.

MIS (Management Information System)

Báo cáo quản lý cho Admin: số học viên, doanh thu, tỉ lệ hoàn thành, bảng điểm tổng hợp.

DSS (Decision Support System)

Hỗ trợ quyết định (ví dụ: phân tích churn/đề xuất học viên cần chăm sóc, tối ưu lịch lớp, đề xuất khóa học nên mở dựa trên dữ liệu).

ESS/EIS (Executive Support)

Dashboard điều hành cho ban lãnh đạo (xu hướng doanh thu, KPI chiến lược).

LMS (Learning Management System) — chuyên biệt (kết hợp TPS + MIS + chức năng học tập): quản lý khóa học, nội dung, theo dõi tiến độ học tập.

3) Mô hình phát triển phần mềm — Agile (Scrum) — đề xuất & lý do

Đề xuất: Áp dụng Agile Scrum (sprint 1–3 tuần) kèm CI/CD.

Lý do :

Nền tảng học trực tuyến cần nhanh phản hồi người dùng (giảng viên + học viên) → yêu cầu dễ thay đổi.

Giao diện, UX, tính năng học trực tuyến thường yêu cầu lặp (thử nghiệm A/B, điều chỉnh).

Scrum cho phép chia module (đăng ký, học, thanh toán, báo cáo) deliver từng phần — có thể triển khai MVP sớm.

Hỗ trợ tích hợp với dịch vụ bên thứ 3 (Zoom, payment) và liên tục test/triển khai bằng CI/CD.

(Nếu project yêu cầu ổn định cực cao/đặc thù pháp lý, có thể hybrid: Agile + gated milestones để kiểm soát rủi ro.)

4) 3 sơ đồ UML sẽ dùng (nếu đi sâu thiết kế)

Use Case Diagram — mô tả chức năng chính theo tác nhân (quick overview cho stakeholders).

Class Diagram (kèm ERD cho CSDL) — mô hình hóa dữ liệu: User, Course, Enrollment, Lesson, Assignment, Grade, Payment.

Sequence Diagram (ví dụ: quy trình “Học viên đăng ký & thanh toán” hoặc “Giảng viên chấm bài”) — mô tả luồng tương tác giữa actor và các thành phần hệ thống.

**Bài 9 :**

1) Tác nhân (Actors) & chức năng chính

Khách hàng (Customer)

Tạo đơn hàng (create order)

Tra cứu trạng thái đơn / theo dõi lộ trình (tracking)

Hủy/đổi thông tin đơn, phản hồi/ khiếu nại

Thanh toán / xem hoá đơn

Nhân viên vận chuyển (Driver / Courier)

Nhận đơn, cập nhật trạng thái (picked up, in-transit, delivered, failed)

Ghi bằng chứng giao hàng (Ảnh, chữ ký)

Báo lỗi giao (delay, địa chỉ sai)

Nhân viên điều phối (Dispatcher / Ops)

Gán đơn cho tài xế, tối ưu tuyến (reassign)

Theo dõi trạng thái thời gian thực, điều phối sự cố

Quản lý khu vực (Area Manager / Supervisor)

Xem báo cáo hiệu suất khu vực, vùng tồn đọng, SLA vi phạm

Quản lý nhân sự khu vực, phân tích nguyên nhân chậm trễ

Admin hệ thống (System Admin)

Quản lý người dùng, phân quyền, cấu hình tuyến/giá, tích hợp đối tác (warehouse, carrier)

CEO / Ban điều hành (Executive)

Xem dashboard tổng quan: giao dịch, vùng hoạt động, tỉ lệ giao đúng hẹn, KPIs

Hệ thống bên thứ ba

GPS/telemetry, cổng thanh toán, kho (WMS), đối tác giao nhận.

2) Phân loại chức năng theo loại hệ thống thông tin

TPS (Transaction Processing System)

Tạo/ghi nhận đơn hàng, thanh toán, cập nhật trạng thái giao hàng, ghi bằng chứng giao hàng.

(Ví dụ: API nhận đơn, mobile app driver cập nhật trạng thái)

MIS (Management Information System)

Báo cáo định kỳ cho quản lý: số đơn/ngày, tỉ lệ hoàn/huỷ, doanh thu theo khu vực, thời gian giao trung bình.

Dashboard trung cấp (supervisor) để giám sát hoạt động hàng ngày.

DSS (Decision Support System)

Phân tích để quyết định: gợi ý tối ưu tuyến, dự báo vùng tồn đọng, what- if (nếu tăng đơn 30% thì cần thêm bao nhiêu tài xế).

Công cụ phân tích nguyên nhân chậm trễ (root-cause) và mô phỏng phân bổ nguồn lực.

EIS / ESS (Executive Information System)

Dashboard cấp cao cho CEO: KPIs chiến lược, heatmap vùng hoạt động, tỉ lệ giao đúng hẹn theo thời gian, doanh thu theo vùng/đối tác.

Lưu ý: Một chức năng cụ thể có thể thuộc nhiều lớp: ví dụ “tracking thời gian thực” là TPS (data stream) nhưng feed dữ liệu cho MIS/DSS/EIS.

3) Chọn mô hình phát triển phần mềm — Agile (Scrum) + DevOps (CI/CD)

Đề xuất: Áp dụng Agile (Scrum) với sprints 1–2 tuần, kết hợp DevOps/CI-CD cho triển khai liên tục và rollout theo khu vực.

Lý do:

Logistic cần tích hợp nhiều hệ thống bên ngoài (GPS, WMS, payment) — yêu cầu lặp, test tích hợp sớm.

Thay đổi yêu cầu cao (tối ưu tuyến, chính sách phí, SLA) → Agile cho phép phản hồi nhanh.

Cần triển khai theo vùng (phased rollout) và cập nhật thường xuyên → DevOps/CI-CD giúp deploy an toàn, rollback nhanh.

Sprint ngắn giúp sớm có MVP (core TPS) để chạy pilot, sau đó thêm MIS/DSS/EIS theo iteration.

Khi nào consider hybrid: nếu có yêu cầu an toàn/tuân thủ rất chặt, dùng Agile nhưng thêm gated milestones (hybrid Agile+Waterfall) cho phần audit/kiểm thử pháp lý.

4) 4 sơ đồ UML đề xuất & mục đích ngắn

Use Case Diagram

Mục đích: Hiển thị actors và các chức năng nghiệp vụ chính (tạo đơn, tracking, cập nhật trạng thái, báo cáo) — dùng cho thảo luận với stakeholders.

Class Diagram (kèm ERD)

Mục đích: Mô hình hóa cấu trúc dữ liệu cốt lõi: Order, Customer, Shipment, Driver, Route, ProofOfDelivery, Invoice — dùng làm cơ sở DB/schema.

Sequence Diagram (ví dụ: “Create Order & Assign Driver”, “Update Delivery Status”)

Mục đích: Mô tả luồng tương tác giữa actor → UI/API → Order Service → Routing Service → Driver App → WMS, rõ thứ tự message và call/response, hữu ích cho thiết kế API.

Activity Diagram (Order Lifecycle) hoặc Deployment Diagram (tùy mục tiêu)

Activity Diagram: mô tả luồng công việc chi tiết cho lifecycle đơn (Created → Picked → In-Transit → Delivered/Failed → Closed), kèm luồng xử lý exception (delay, reassign).

Deployment Diagram (nếu cần): mô tả kiến trúc triển khai: mobile apps, backend services, message broker, CDN, DB cluster, edge nodes — hữu ích cho planning hạ tầng & DevOps.