**Bài 1:**

* Giao dịch bán hàng: TPS dùng để ghi lại, xử lý, và lưu trữ các giao dịch hàng ngày của doanh nghiệp
* Phân tích xu hướng kinh doanh: DSS giúp **phân tích dữ liệu, mô phỏng, và hỗ trợ ra quyết định** dựa trên các mô hình hoặc truy vấn linh hoạt
* Bảng tổng quan hiệu suất hàng tháng dành cho CEO: EIS cung cấp **thông tin tổng hợp, trực quan, chiến lược** giúp CEO theo dõi **hiệu suất tổ chức** và ra quyết định nhanh

**Bài 2:**

1. Waterlfall : vì yêu cầu rõ ràng, ít thay đổi
2. Agile: vì cần linh hoạt, phản hồi nhanh với người dùng
3. Spiral: vì phức tạp, rủi ro cao, cần đánh giá và kiểm soát chặt chẽ

**Bài 3:**

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| Con người | -Khách hàng  -Tài xế giao hàng  -Khách hàng nhận món ăn  -Nhà hàng  -Quản trị viên |
| Dữ liệu | -Thông tin người đặt  -Thông tin món ăn  -Thông tin chi tiết đơn hàng  -Địa chỉ nhận hàng  -Dữ liệu tài xế |
| Quy trình | -Đặt món  -Xác nhận  -Giao hàng  -Thanh toán  -Đánh giá |
| Phần mềm | -Ứng dụng đặt món  -Hệ thống quản lý  -Điều phối tài xế  -Trang riêng tương tác khách hàng |
| Phần cứng | -Thiết bị người dùng, internet  -Điện thoại  -Máy tính |

**Bài 4:**

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án "Ứng dụng điểm danh"** |
| Planning | -Xác định mục tiêu và phạm vi dự án  -Lập kế hoạch thời gian nhân sự  -Phân chia công việc cho các thành viên trong nhóm |
| Analysis | -Thu nhập yêu cầu từ giảng viên, sinh viên, phòng đào tạo  -Xác định chức năg chính: đăng nhập, điểm danh, xem lịch sử, xuất báo cáo |
| Design | -Thiết kế giao diện ứng dụng (web/mobile)  -Thiết kế cơ sở dữ liệu: bảng SinhVien, LopHoc, DiemDanh  -Xác định luồng xử lý dữ liệu và bảo mật |
| Implementation | -Viết mã chương trình cho các chức năng chính  -Kết nối ứng dụng với CSDL  -Xây dựng API cho điểm danh và báo cáo |
| Testing | -Kiểm tra tính đúng đắn của các chức năng -Thử điểm danh thực tế, kiểm tra đăng nhập, báo cáo -Phát hiện và sửa lỗi trước khi phát hành |
| Deployment & Maintenance | -Đưa ứng dụng vào sử dụng trong trường học -Hướng dẫn người dùng (giảng viên, sinh viên) -Theo dõi hoạt động, sửa lỗi, cập nhật và thêm tính năng mới |

**Bài 5:**

1. Planning
   * Xác định mục tiêu: Giúp giảng viên điểm danh nhanh, tránh gian lận, tự động lưu kết quả
   * Xác định người dùng chính: Giảng viên, Sinh viên, Phòng đào tạo
   * Lên kế hoạch thời gian, nhân sự, chi phí tạo ứng dụng
   * Dự đoán rủi ro: Lỗi mạng, lỗi QR, điểm danh hộ
2. Analysis
   * Thu thập nhu cầu của từng nhóm
   * Xác định chức năng chính đăng nhập, tạo buổi học, quét QR, lưu dữ liệu, báo cáo
3. Design
   * Thiết kế giao diện đơn giản cho giảng viên và sinh viên
   * Thiết kế cơ sở dữ liệu
   * Thiết kế cách hoạt động
   * Xác định các phần mềm và thiết bị cần dùng

**Bài 6:**

|  |  |
| --- | --- |
| **Tình huống** | **Sơ đồ UML phù hợp** |
| A. Mô tả chức năng người dùng có thể thực hiện trên ứng dụng học tiếng Anh | Use Case Diagram |
| B. Mô tả lớp NguoiDung, KhoaHoc, BaiHoc và quan hệ giữa chúng | Class Diagram |
| C. Mô tả luồng học viên bắt đầu → vào học → làm bài → hoàn thành | Activity Diagram |
| D. Mô tả cách hệ thống triển khai trên các máy chủ, thiết bị | Deployment Diagram |
| E. Mô tả thứ tự tương tác giữa học viên và hệ thống khi nộp bài | Sequence Diagram |

**Bài 7:**

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm** |
| Planning | -Xác định mục tiêu: đăng ký tiêm online, xếp lịch, theo dõi đã/chưa tiêm  -Xác định người dùng: ng dân, nhân viên, quản trị  -Lập kế hoạch thời gian, nhân sự, rủi ro |
| Analysis | -Thu nhập yêu cầu: đăng ký, chọn lịch, đổi/huỷ, xem danh sách  -Xác định chức năg cho từng vai trò  -Đề ra yêu cầu phi chức năng: bảo mật, tốc độ, dễ dùng |
| Design | -Thiết kế giao diện: form đăng ký, chọn lịch, trang quản lý -Thiết kế cơ sở dữ liệu: NgườiDân, LịchHẹn, CơSở, TiêmChủng -Mô tả luồng: đăng ký → xác nhận → tiêm → theo dõi |
| Implementation | -Viết mã chương trình cho các chức năng chính  -Tạo API đăng ký, xác nhận, quản lý danh sách -Thiết lập phân quyền và bảo mật đăng nhập |
| Testing | -Kiểm tra chức năng, tốc độ và tính bảo mật -Thử nghiệm với nhân viên trước khi sử dụng thật |
| Deployment & Maintenance | -Đưa hệ thống vào hoạt động thật -Theo dõi, sửa lỗi, cập nhật và cải tiến khi có phản hồi |

**Bài 8:**

1. **Tác nhân chính và chức năng tương   
   Học viên:**
   * Đăng ký tài khoản/ ghi danh khoá học, thanh toán
   * Tham gia học, xem điểm/chứng chỉ

**Giảng viên:**

* + Tạo/quản lý kháo học và bài học. giao bài/đề thi
  + Chấm điểm, phản hỏi, nhận xét, điểm danh

**Admin**

* + Quản lý người dùng và phân quyền
  + Quản lý lịch khai giảng, học phí, lớp học
  + Theo dõi báo cáo/thống kê

**Cổng thanh toán:** xử lý giao dịch

**Dịch vụ thông báo:** (Email/SMS/Zalo) gửi xác nhận, nhắc lịch học/thi

1. **Phân loại hệ thống thông tin phù hợp**

**TPS:** Đăng ký/ghi danh, thanh toán, nộp bài, ghi nhận điểm → các giao dịch hằng ngày

**MIS:** Báo cáo định kỳ số học viên theo lớp/khoá/tháng, sổ điểm, tỉ lệ hoàn thành

**DSS:** Phân tích & dự báo dự báo nhu cầu mở lớp, nhận diện học viên nguy cơ rớt, tối ưu lịch/giảng viên

**EIS:** Dashboard KPI cấp cao doanh thu, tăng trưởng, CAC, NPS, tỉ lệ chuyển đổi

1. **Mô hình phát triển phần mềm khuyến nghị**

Mô hình **Agile** phù hợp nhất vì:

* + Nền tảng học trực tuyến thay đổi thường xuyên theo phản hồi học viên – giảng viên; nhiều tích hợp (thanh toán, live class), cần cập nhật phiên bản mới và cải tiến liên tục
  + Triển khai cập nhật 2 tuần 1 lần; ưu tiên các “miếng” giá trị: ghi danh + thanh toán → học và nộp bài → chấm điểm → báo cáo

1. **Ba sơ đồ UML dùng khi thiết kế sâu**
   1. **Use Case Diagram** – ai làm gì: Học viên/ Giảng viên/ Admin/ Payment/ Notification và các ca sử dụng tương ứng.
   2. **Class Diagram** – dữ liệu & quan hệ: HocVien, KhoaHoc, BaiHoc, Lop, DangKy, ThanhToan, BaiTap, BaiNop, Diem, Role, Permission.
   3. **Sequence Diagram** – thứ tự tương tác:

Luồng **ghi danh + thanh toán,**

Luồng **nộp bài → chấm điểm → trả kết quả,**

Luồng **Admin xem báo cáo.**

**Bài 9:**

1. **Tác nhân (actors) và chức năng chính**

**Kách hàng**

* + Tạo đơn hàng (nhập người gửi/nhận, hàng hóa, COD, dịch vụ)
  + Theo dõi trạng thái giao hàng, nhận thông báo
  + Yêu cầu đổi địa chỉ/ghi chú, khiếu nại

**Shipper**

* + Nhận việc, quét mã kiện, cập nhật trạng thái (nhận, đang giao, giao thành công/thất bại)
  + Báo cáo sự cố, đề xuất tuyến

**Điều phối/ Quản lý vận hành**

* + Phân tuyến, gán đơn, theo dõi hiệu suất theo ca/khu vực
  + Xem tồn đọng, tắc nghẽn theo trạm/khu vực; xuất báo cáo định kỳ

**CEO:** Xem tổng quan: doanh thu/giao dịch, vùng hoạt động, tỉ lệ giao đúng hẹn (OTD/OTP), KPI chiến lược

**Cổng thanh toán/Đối tác vận tải:** Xử lý giao dịch COD/online; nhận/đẩy sự kiện liên thông

1. **Phân loại từng chức năng vào hệ thống thông tin phù hợp**
   * Khách hàng tạo đơn, theo dõi trạng thái: **TPS**
   * Shipper cập nhật trạng thái giao hàng: **TPS**
   * Quản lý xem báo cáo hiệu suất, tồn đọng theo khu vực (định kỳ): **MIS**
   * Quản lý phân tích tắc nghẽn, tối ưu tuyến, dự báo khối lượng: **DSS**
   * CEO dashboard tổng quan KPI, vùng hoạt động, OTD: **EIS**
2. **Mô hình phát triển phần mềm phù hợp**

Nên dùng mô hình **Agile (Scrum)** kết hợp **hướng sự kiện (event-driven)** cho phần vận hành thời gian thực

* + Dùng **Agile** vì nghiệp vụ logistics thay đổi liên tục (chính sách COD, giá cước), cần ra mắt phiên bản mới sớm và cải tiến theo phản hồi từ người dùng
  + Dùng **event-driven** vì trạng thái đơn hàng thay đổi liên tục; kiến trúc dựa trên **sự kiện trạng thái** giúp cập nhật thời gian thực

1. **4 sơ đồ UML đề xuất**
2. **Use Case Diagram** – *Ai làm gì với hệ thống*
   * Actors: Khách hàng, Shipper, Điều phối, CEO, Cổng thanh toán.
   * Use cases: tạo đơn, theo dõi, cập nhật trạng thái, phân tuyến, xem báo cáo, xem KPI.
3. **Activity Diagram** –*Quy trình nghiệp vụ đầu–cuối*
   * Luồng vòng đời đơn: Tạo đơn → nhập kho → liên trạm → giao → giao thành công/ thất bại → xử lý lại/hoàn; rẽ nhánh theo sự cố.
4. **Sequence Diagram** –*Thứ tự tương tác thời gian thực*
   * Case “Shipper cập nhật trạng thái”: App Shipper → API Gateway → Order Service → Event Bus → Tracking Service → Notification → cập nhật khách hàng.
   1. **Class Diagram** – *Mô hình dữ liệu & quan hệ*
   * Lớp tiêu biểu: Order, Parcel, Shipment, Stop, StatusEvent, Customer, Courier, Route, Hub, Invoice, quan hệ 1-n giữa Order–StatusEvent, Shipment–Stop.