**SESSION 01: TỔNG QUAN VỀ PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

BÀI 1:

- Giao dịch bán hàng: **loại hệ thống thông tin tương ứng là TPS**

- Phân tích xu hướng kinh doanh: **DSS**

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

BÀI 2:

* Dự án A: **Waterfall** -> phù hợp khi yêu cầu đã rõ ràng, dự án nhỏ, và có thể thực hiện theo trình tự cố định.
* Dự án B: **Agile** -> phù hợp vì dự án cần thích ứng nhanh, cập nhật thường xuyên và tương tác chặt chẽ với người dùng.
* Dự án C: **Spiral** -> phù hợp vì cần kiểm soát rủi ro, bảo mật, và có khả năng điều chỉnh qua từng vòng phát triển.

BÀI 3:

|  |  |
| --- | --- |
| Thành phần | Mô tả |
| Con người | Người quản lí, khách hàng, tài xế, |
| Dữ liệu | Thông tin người dùng, tt món ăn, tt tài xế |
| Quy trình | Đặt món – giao hàng – thanh toán |
| Phần mềm | App, Website đặt hàng, quản lí món |
| Phần cứng | Thiết bị người dùng, Máy in hóa đơn |

BÀI 4:

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án "Ứng dụng điểm danh"** |
| Planning | Xác định mục tiêu, phạm vi, thời gian và phân công công việc. |
| Analysis | Thu thập, phân tích yêu cầu từ giảng viên và sinh viên. |
| Design | Thiết kế giao diện, cơ sở dữ liệu và kiến trúc hệ thống. |
| Implementation | Lập trình các chức năng như điểm danh, đăng nhập, báo cáo. |
| Testing | Kiểm thử chức năng, sửa lỗi và đảm bảo hệ thống ổn định. |
| Deployment & Maintenance | Triển khai ứng dụng, cập nhật và bảo trì khi cần. |

BÀI 5:

**1. Planning (Lập kế hoạch)**

* **Mục tiêu:** Xây dựng hệ thống điểm danh bằng QR code.
* **Phạm vi:**
  + Giảng viên tạo buổi học và mã QR.
  + Sinh viên quét mã để điểm danh.
  + Phòng đào tạo xem thống kê.
* **Công nghệ:** ReactJS, NodeJS, MongoDB.
* **Thời gian:** 8 tuần, 5 thành viên.

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

* **Actors:** Giảng viên, Sinh viên, Phòng đào tạo.
* **Chức năng chính:**
  1. Giảng viên tạo buổi học, xem danh sách điểm danh.
  2. Sinh viên đăng nhập, quét QR điểm danh.
  3. Phòng đào tạo xem báo cáo thống kê.

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

* **Sơ đồ UML:** *Use Case Diagram*
* **Thiết kế:**
  + **Frontend:** giao diện web/app quét mã.
  + **Backend:** xử lý tạo QR, ghi nhận điểm danh.
  + **Database:** User, Class, Session, Attendance.

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:

**1. Planning (Lập kế hoạch)**

* Xác định mục tiêu: tạo hệ thống đăng ký tiêm online.
* Phạm vi: người dân đăng ký → trung tâm xác nhận, xếp lịch → theo dõi danh sách tiêm.
* Lên kế hoạch nhân sự, thời gian, công nghệ (web + database).

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

* **Người dùng:** Người dân, nhân viên trung tâm, quản lý.
* **Chức năng chính:**
  1. Người dân đăng ký tiêm.
  2. Trung tâm xác nhận & xếp lịch.
  3. Quản lý theo dõi tình trạng tiêm.
* **Phi chức năng:** bảo mật, dễ dùng, hoạt động ổn định.

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

* Vẽ **Use Case**, **Database**, **Giao diện**.
* Thiết kế luồng: đăng ký → xác nhận → xếp lịch → cập nhật kết quả.
* Xây dựng mô hình dữ liệu: User, Schedule, Vaccine, Appointment.

**4. Implementation (Xây dựng)**

* Lập trình frontend + backend theo thiết kế.
* Kết nối CSDL, viết API đăng ký và quản lý lịch tiêm.
* Kiểm tra chức năng từng phần (unit test).

**5. Testing (Kiểm thử)**

* Test chức năng: đăng ký, xác nhận, hiển thị danh sách.
* Test bảo mật, tốc độ, giao diện.
* Sửa lỗi trước khi triển khai.

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

* Đưa hệ thống lên server.
* Theo dõi, hỗ trợ người dùng, cập nhật tính năng.
* Sao lưu dữ liệu và khắc phục sự cố.

BÀI 8:

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

|  |  |
| --- | --- |
| **Tác nhân** | Chức năng chính của hệ thống |
| Học viên | - Đăng ký tài khoản, đăng nhập hệ thống.  - Xem và đăng ký khóa học.  - Học trực tuyến (xem bài, làm bài, nộp bài).  - Xem điểm và tiến độ học tập. |
| Giảng viên | - Tạo, chỉnh sửa và quản lý bài học.  - Chấm điểm, đánh giá học viên.  - Theo dõi tiến độ học viên trong khóa học. |
| Admin (Quản trị) | - Quản lý tài khoản người dùng (học viên, giảng viên).  - Phân quyền truy cập hệ thống.  - Xem báo cáo, thống kê số lượng học viên, khóa học, kết quả học tập.  - Quản lý dữ liệu toàn hệ thống. |

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

* Hệ thống này **thuộc nhóm MIS (Management Information System)** — **Hệ thống thông tin quản lý.**
* **Giải thích:**
  + Hệ thống không chỉ xử lý giao dịch (đăng ký, học, chấm điểm – thuộc TPS),
  + mà còn **tổng hợp dữ liệu, tạo báo cáo, hỗ trợ quản lý ra quyết định** (ví dụ: thống kê tỉ lệ hoàn thành khóa, doanh thu, kết quả học viên).  
    → Vì vậy, đây là **MIS**, có thể tích hợp thêm **TPS** ở mức xử lý dữ liệu cơ bản.

### ****3. Mô hình phát triển phần mềm phù hợp nhất****

* **Đề xuất:** Mô hình **Agile (phát triển linh hoạt)** hoặc **Iterative (phát triển lặp)**.

**Lý do chọn:**

1. Dự án có nhiều chức năng khác nhau (học viên, giảng viên, admin), nên cần phát triển từng phần theo vòng lặp.
2. Dễ dàng **thay đổi, mở rộng chức năng** dựa trên phản hồi thực tế từ người dùng.
3. Cho phép **demo sớm từng phiên bản**, giúp khách hàng (trung tâm ngoại ngữ) kiểm tra và góp ý kịp thời.
4. Giảm rủi ro so với mô hình Waterfall vì có thể điều chỉnh giữa chừng mà không ảnh hưởng toàn bộ hệ thống.

Ví dụ:

* Sprint 1: Xây chức năng học viên đăng ký và tham gia khóa học.
* Sprint 2: Thêm phần quản lý bài học cho giảng viên.
* Sprint 3: Hoàn thiện phần báo cáo và phân quyền cho admin.

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

1. **Use Case Diagram (Sơ đồ ca sử dụng):**
   * Mô tả các tác nhân (học viên, giảng viên, admin) và các chức năng tương tác với hệ thống.
   * Giúp hiểu tổng quan hệ thống làm gì và ai sẽ dùng.
2. **Class Diagram (Sơ đồ lớp):**
   * Mô tả các lớp chính như user, courscùng mối quan hệ giữa chúng.
   * Giúp lập trình viên thiết kế CSDL và cấu trúc code logic.
3. **Sequence Diagram (Sơ đồ tuần tự):**
   * Mô tả trình tự tương tác giữa các đối tượng, ví dụ:
     + Học viên đăng nhập → chọn khóa học → hệ thống xác nhận → thêm vào danh sách học.
   * Giúp hiểu rõ luồng xử lý từng chức năng.

BÀI 9:

**1) Các tác nhân (Actors) và chức năng chính**

* **Khách hàng**
  + Tạo đơn hàng (địa chỉ, kích thước, loại hàng).
  + Theo dõi trạng thái đơn (pickup, in-transit, delivered, exception).
  + Hủy/điều chỉnh đơn (nếu cho phép).
  + Nhận thông báo (SMS/email).
* **Nhân viên vận chuyển (Driver / Courier)**
  + Nhận/nhận lệnh giao hàng.
  + Cập nhật trạng thái theo thời gian thực (đã lấy hàng, đang giao, giao thành công, giao thất bại).
  + Ghi chú / chụp ảnh bằng chứng giao hàng.
* **Quản lý vận hành (Ops Manager / Area Manager)**
  + Xem danh sách đơn theo trạng thái/khu vực.
  + Phân công/điều phối nhân viên giao.
  + Theo dõi KPIs: tỉ lệ giao đúng hạn, đơn tồn, thời gian trung bình giao.
* **CEO / Ban giám đốc**
  + Xem dashboard tổng quan: số lượng giao dịch, doanh thu, vùng hoạt động, tỉ lệ giao đúng hạn, trend theo thời gian.
  + Lấy báo cáo chiến lược và chỉ số quyết
* **Admin hệ thống**
  + Quản lý người dùng, dịch vụ, cấu hình vùng giá, cước phí.
  + Xử lý exceptions (hoàn tiền, khiếu nại).

2) Phân loại từng chức năng vào hệ thống thông tin phù hợp

|  |  |  |
| --- | --- | --- |
| **Chức năng** | Loại hệ thống phù hợp | Lí do ngắn gọn |
| Tạo đơn, cập nhật trạng thái (real-time) | TPS | Xử lý giao dịch hàng ngày, cần tin cậy và nhất quán. |
| Theo dõi đơn cho khách hàng | TPS + MIS | TPS lưu và trả trạng thái; MIS tóm tắt lịch sử, báo cáo cho khách nếu cần. |
| Phân công/điều phối nhân viên | DSS | Cần phân tích dữ liệu (độ ưu tiên, tuyến đường, tải) để ra quyết định tối ưu. |
| Báo cáo vận hành theo khu vực, KPIs | MIS | Tổng hợp dữ liệu, báo cáo cho quản lý vận hành. |
| Dashboard CEO: tổng quan, xu hướng, tỉ lệ đúng hạn | EIS | Giao diện cho lãnh đạo, tập trung chỉ số cấp cao và xu hướng. |
| Phân tích nguyên nhân chậm/hỏng hàng, mô phỏng tác động đổi tuyến | DSS | Hỗ trợ mô phỏng, what-if, tối ưu hóa |

**3) Chọn mô hình phát triển phần mềm & giải thích**

**Đề xuất:** **Agile (Scrum)** kết hợp **Iterative delivery**; kiến trúc microservices khi đi vào triển khai.

**Lý do:**

* Dự án có nhiều stakeholders (khách hàng, nhân viên giao, ops, CEO) với yêu cầu thay đổi theo thực tế → **Agile** cho phép phản hồi liên tục và điều chỉnh nhanh.
* Tách chức năng (orders, tracking, routing, reporting) ra từng module/sprint giúp giao sớm các tính năng giá trị (MVP).
* Microservices/Modular giúp:
  + Mở rộng từng phần (ví dụ: service tracking vs. service reporting).
  + Triển khai độc lập, scale cho phần TPS (tracking) khi có tải cao.
* Hỗ trợ tích hợp với bên thứ ba (map API, SMS gateway) dễ dàng theo từng sprint.

**Sprint gợi ý:**

* Sprint 1: MVP Orders + Tracking (khách + driver).
* Sprint 2: Notifications + Basic Ops dashboard.
* Sprint 3: Routing/assignment + DSS cơ bản.
* Sprint 4: EIS / báo cáo cho CEO + tối ưu hiệu suất.

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

1. **Use Case Diagram**
   * **Mục đích:** Mô tả ai làm gì (actors và chức năng chính) — hữu ích cho thu thập yêu cầu và thảo luận với stakeholders.
2. **Class Diagram (Sơ đồ lớp)**
   * **Mục đích:** Thiết kế mô hình dữ liệu / cấu trúc đối tượng: Customer, Order, Shipment, Driver, Route, Status, Region, KPIReport. Dùng để thiết kế DB và lớp trong code.
3. **Sequence Diagram (Sơ đồ tuần tự)**
   * **Mục đích:** Mô tả trình tự tương tác cho các luồng quan trọng, ví dụ: Khách tạo đơn → Hệ thống lưu → Dispatch service phân tuyến → Driver nhận → Driver cập nhật trạng thái → Hệ thống notify khách. Rõ ràng các message/API gọi giữa services.
4. **Activity Diagram (Sơ đồ hoạt động) hoặc Deployment Diagram** (chọn 1, tùy mục tiêu)
   * **Activity Diagram (nếu cần phân tích quy trình nghiệp vụ):** Mô tả luồng nghiệp vụ phức tạp như lifecycle của đơn (tạo → pick-up → in-transit → delivery → exception → close). Hữu ích cho QA và mô tả luồng xử lý.
   * **Deployment Diagram (nếu cần thiết kế hạ tầng):** Mô tả triển khai trên server, message broker, mobile apps, GPS devices — hữu ích cho DevOps/arquitect.