Bài 1:

Loại hệ thống thông tin tương ứng của từng chức năng sau:

* 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 :

| **Dự án** | **Mô tả ngắn** |
| --- | --- |
| Waterfall | **Phần mềm quản lý điểm cho trường cấp 2, yêu cầu rõ ràng** |
| Agile | **Ứng dụng mobile đặt lịch khám bệnh, yêu cầu linh hoạt** |
| Spiral | **Hệ thống ngân hàng điện tử có bảo mật và độ phức tạp cao** |

Bài 3:

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| Con người | Khách hàng : Đặt đồ ăn , Shipper: Giao hàng, Người bán đồ ăn: Làm đồ ăn |
| Dữ liệu | Địa chỉ giao hàng và thông tin người đặt hàng, thông tin người giao hàng, thông tin và địa chỉ quán ăn |
| Quy trình | Khách hàng đặt đơn hàng -> Người bán xác nhận -> Thông tin được gửi đến shipper -> Shipper nhận đơn và giao đến khách hàng ->Khách hàng nhận hàng và thanh toán |
| Phần mềm | Ứng dụng đặt đồ ăn cho khách hàng, ứng dụng nhận đơn cho tài xế, ứng dụng nhận đơn cho và gán tài xế nhà hàng, cơ sở dữ liệu,web admin |
| Phần cứng | Máy chủ, thiết bị đặt hàng, thiết bị giao hàng, thiết bị nhận đơn hàng |

Bài 4:

Bạn đang phát triển một ứng dụng điểm danh sinh viên trong trường học. Hãy liệt kê 6 giai đoạn SDLC và mô tả ngắn bạn sẽ làm gì trong từng giai đoạn đó.

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án "Ứng dụng điểm danh"** |
| Planning | Xác định mục tiêu: (Ứng dụng điểm danh cho sinh viên trong trường) , phạm vi: (web, mobile) |
| Analysis | Người tham gia: Giáo viên, sinh viên, admin, phòng đào tạo  Phân tích yêu cầu: Điểm danh sinh viên theo lớp/ buổi , phương thức điểm danh: app, web,QR  Phân tích chức năng: Thêm lớp học , thêm buổi học, thêm sinh viên, chức năng điểm danh, xử lý điểm danh muộn và vắng, báo cáo cho giảng viên, gửi báo cáo, cập nhật trạng thái điểm danh |
| Design | Thiết kế cấu trúc: client-API-DB  Dữ liệu: Lớp học, sinh viên, buổi học, trạng thái điểm danh,  Giao diện ứng dựng: giao diện điểm danh, giao diện quản lí của giảng viên, giao diện sinh viên, giao diện quản lí admin, |
| Implementation | Phát triển FE,BE,API, tích hợp ứng dụng thứ 3 (vd: QR), cấu hình Database |
| Testing | Sinh viên, giảng viên, phòng đào tạo, tester dùng và kiểm thử hiệu năng, bảo mật về quyền truy cập của sinh viên, giảng viên và admin, chống gian lận |
| Deployment & Maintenance | Triển khai ứng dụng,hướng dẫn sử dụng, bảo trì, sửa lỗi và nâng cấp, tối ưu hiệu năng và thêm tính năng mới |

Bài 5:

Cho các tình huống sau. Hãy chọn sơ đồ UML tương ứng:

Một trường đại học muốn xây dựng **hệ thống điểm danh sinh viên bằng QR code**. Hệ thống cho phép:

* Giảng viên tạo buổi học và quét QR
* Sinh viên dùng ứng dụng quét mã để điểm danh
* Phòng đào tạo theo dõi thống kê điểm danh

**Yêu cầu:**  
Hãy trình bày bạn sẽ thực hiện **3 giai đoạn đầu của SDLC** như thế nào nếu là thành viên nhóm phát triển phần mềm này:

1. Planning
2. Requirement Analysis
3. System Design

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án "Ứng dụng điểm danh"** |
| Planning | Xác định mục tiêu: (Ứng dụng điểm danh cho sinh viên trong trường) , phạm vi: (web, mobile), phương thức điểm danh: mã QR |
| Requirement Analysis | Use cases chính:  Giảng viên: tạo buổi học, phát QR, xem/sửa điểm danh, xuất báo cáo.  Sinh viên: đăng nhập, quét QR, xác nhận check‑in.  Phòng đào tạo: xem thống kê theo lớp/khoa/thời gian, xuất tổng hợp.  Phân tích yêu cầu: Điểm danh sinh viên theo lớp/ buổi , phương thức điểm danh: QR  Phân tích chức năng: thêm lớp học , thêm sinh viên, chức năng điểm danh, xử lý điểm danh muộn và vắng, báo cáo cho giảng viên, gửi báo cáo, phòng đào tạo theo dõi thống kê điểm danh của các lớp học , giảng viên tạo buổi học , cập nhật trạng thái điểm danh |
| System Design | Thiết kế cấu trúc: client-API-DB  Dữ liệu: Lớp học, sinh viên, buổi học, trạng thái điểm danh, mã QR điểm danh và bảo mật  Giao diện ứng dựng: giao diện điểm danh, giao diện quản lí của giảng viên, giao diện sinh viên, giao diện quản lí admin, |

Bài 6:

Cho các tình huống sau. Hãy chọn sơ đồ UML tương ứng:

|  |  |
| --- | --- |
| **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:

Bạn được yêu cầu tham gia phát triển **hệ thống hỗ trợ đăng ký tiêm chủng online** cho một trung tâm y tế.

Chức năng hệ thống bao gồm:

* Người dân đăng ký tiêm
* Trung tâm xác nhận và xếp lịch tiêm
* Quản lý theo dõi danh sách người đã tiêm và chưa tiêm

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án** |
| Planning | Xác định mục tiêu: hệ thống hỗ trợ đăng ký tiêm chủng cho người dùng tại một trung tâm y tế  Phạm vi: (app,web) |
| Analysis | Use cases chính:  Người đăng ký tiêm: có thể đăng ký tiêm và nhận thông báo khi được xếp lịch, theo dõi được tất cả thông tin và lịch sử tiêm của mình  Trung tâm :xác nhận yêu cầu đăng ký và xếp lịch cho khách hàng, gửi thông báo cho khách hàng, có thế xem được tất cả thông tin và lịch sử tiêm của tất cả khách hàng đã sử dụng hệ thống, quản lí và theo dõi được danh sách người đã tiêm và chưa tiêm  Chức năng chính: Đăng nhập và đăng ký, Đăng ký tiêm, Xác nhận yêu cầu đăng ký, Xếp lịch cho người đăng ký , quản lí danh sách khách hàng và trạng thái tiêm, chức năng thay đổi, chỉnh sửa thông tin và trạng thái khách hàng |
| Design | Thiết kế Cấu trúc: client- API- DB  Dữ liệu: Người dùng , Trung tâm y tế, trạng thái tiêm của người dùng  UI: ứng dụng đăng ký tiêm của người dùng, ứng dụng xác nhận đăng ký , xếp lịch và thông báo cho người dùng của trung tâm y tế |
| Implementation | Xây dựng **frontend** (ứng dụng web/app):  +Giao diện người dân đăng ký tiêm, xem lịch, nhận thông báo.  +Giao diện cho trung tâm xác nhận đăng ký, xếp lịch, quản lý danh sách người tiêm.  Xây dựng **backend API**:  +Xử lý yêu cầu đăng ký, xác nhận, và gửi thông báo.  +Quản lý dữ liệu người dùng, lịch tiêm, trạng thái tiêm.  Thiết kế và triển khai **database** (MySQL, MongoDB, v.v) lưu trữ người dùng, trung tâm, lịch sử tiêm.  Kết nối **client – API – DB**, đảm bảo dữ liệu đồng bộ, bảo mật và hoạt động ổn định. |
| Testing | **Việc cần làm trong dự án:**  Kiểm thử **chức năng** :  Đăng ký, đăng nhập, xác nhận, xếp lịch, gửi thông báo, cập nhật trạng thái.Kiểm thử **hiệu năng (Performance testing)**: tốc độ phản hồi, tải người dùng đồng thời.  Kiểm thử **bảo mật :**xác thực, phân quyền, bảo vệ dữ liệu cá nhân.  Kiểm thử **giao diện :**tính dễ dùng, hiển thị đúng, thân thiện người dùng.  Sửa lỗi (debug), tối ưu hệ thống trước khi đưa vào vận hành. |
| Deployment & Maintenance | Triển khai hệ thống lên **server thực tế**  Theo dõi và **bảo trì định kỳ**:  Cập nhật chức năng mới theo nhu cầu thực tế.  Vá lỗi, cải thiện hiệu năng.  Quản lý và sao lưu dữ liệu người dùng, lịch tiêm.  Hỗ trợ kỹ thuật cho trung tâm y tế và người dùng. |

Bài 8:

|  |  |
| --- | --- |
| Các tác nhân | Chức năng |
| Học viên | Đăng ký tài khoản / đăng nhập  - Xem danh sách khóa học  - Đăng ký và tham gia khóa học  - Làm bài kiểm tra, xem điểm, tiến độ học |
| Giảng viên | Tạo và quản lý khóa học, bài học, tài liệu  - Quản lý điểm và nhận xét học viên  - Theo dõi tiến độ học viên  - Tương tác, trả lời câu hỏi học viên |
| Admin | Quản lý người dùng (học viên, giảng viên)  - Phân quyền tài khoản  - Theo dõi báo cáo, thống kê hoạt động |

**Hệ thống này thuộc loại:**  
 **TPS (Transaction Processing System)** – Hệ thống xử lý giao dịch  
vì nó xử lý các giao dịch thường xuyên như: đăng ký học, lưu điểm, cập nhật dữ liệu người dùng.

**Đề xuất:** **Mô hình Agile (Scrum)**

**Lý do chọn:**

* Dự án có **nhiều bên tham gia** (học viên, giảng viên, admin) ->dễ thay đổi yêu cầu trong quá trình làm.
* Agile cho phép **phát triển theo từng giai đoạn nhỏ (Sprint)** ->test và nhận phản hồi nhanh từ trung tâm ngoại ngữ.
* **Cập nhật tính năng linh hoạt**, phù hợp với hệ thống học trực tuyến thường xuyên nâng cấp (thêm khóa học, module mới).
* Đảm bảo **chất lượng UI/UX cao**, vì có thể test người dùng sớm.

| **Tên sơ đồ UML** | **Mục đích sử dụng** |
| --- | --- |
| **Use Case Diagram** | Xác định tác nhân và các chức năng chính của hệ thống (ai làm gì). |
| **Class Diagram** | Mô tả cấu trúc dữ liệu, các lớp như HọcViên, GiảngViên, KhóaHọc, BàiHọc, Điểm, v.v. |
| **Sequence Diagram** | Minh họa luồng tương tác giữa các đối tượng (ví dụ: Học viên đăng ký khóa học → Hệ thống → CSDL). |

Bài 9:

1. CÁC TÁC NHÂN VÀ CHỨC NĂNG CHÍNH

Khách hàng:

+Tạo đơn hàng

+Theo dõi trạng thái giao

+Hủy/điều chỉnh đơn

+Nhận thông báo

+Đánh giá dịch vụ, gửi khiếu nại.

Nhân viên vận chuyển :

+Nhận đơn giao .

+Cập nhật trạng thái

Quản lý/Trưởng khu vực:

+Giám sát hiệu suất đội giao

+Quản lý khiếu nại, hoàn tiền.

CEO / Ban giám đốc:

+Xem dashboard tổng quan: tổng giao dịch, doanh thu, tỷ lệ giao đúng hẹn, vùng hoạt động

+Ra quyết định chiến lược dựa trên báo cáo.

1. PHÂN LOẠI CHỨC NĂNG THEO HỆ THỐNG THÔNG TIN (TPS, MIS, DSS, EIS)

* TPS (Transaction Processing System) — xử lý giao dịch hàng ngày, transactional:

+Tạo đơn hàng.

+Cập nhật trạng thái giao (shipper cập nhật).

+Thanh toán và hoàn tiền.

+Scan POD / lưu bằng chứng giao.

+Notifications (sms/email/push) cơ bản.

* MIS (Management Information System) — báo cáo, tổng hợp cho quản lý:

+Báo cáo số lượng đơn, doanh thu theo ngày/tuần/tháng.

+Thống kê hiệu suất nhân viên / khu vực.

+Báo lỗi/tồn đọng hàng.

* DSS (Decision Support System) — hỗ trợ ra quyết định phức tạp/đề xuất:

+Phân tích nguyên nhân tồn đọng, đề xuất phân bổ lại tuyến.

+Mô phỏng thay đổi số lượng shipper và tác động tới on-time rate.

+Gợi ý tối ưu tuyến (kết hợp dữ liệu bản đồ, traffic).

* EIS (Executive Information System) — dashboard cấp cao cho lãnh đạo:

+Dashboard CEO: KPIs (tỷ lệ giao đúng hẹn, growth, revenue by region).

+Cảnh báo chiến lược (region underperforming, spike of complaints).

Đề xuất: MÔ HÌNH AGILE

Lý do:

+Dự án có nhiều yêu cầu dễ thay đổi theo feedback thực tế.

+Cần phát hành sớm các chức năng cốt lõi rồi mở rộng dần

+Nhận phản hồi từ vận hành thực tế, sửa nhanh.

+Hệ thống logistics cần thử nghiệm ở thực tế để tinh chỉnh

4 SƠ ĐỒ UML ĐỀ XUẤT

* Use Case Diagram

+Mục đích: Xác định actors và các chức năng chính hệ thống hỗ trợ

* Class Diagram

+Mục đích: Mô tả cấu trúc dữ liệu, các lớp / thực thể chính và quan hệ giữa chúng.

+Lớp mẫu: Order, Customer, Shipment, Shipper.

* Sequence Diagram

+Mục đích: Minh hoạ luồng tương tác theo thời gian giữa actors và hệ thống cho các use case quan trọng.

* Activity Diagram (hoặc State Machine Diagram cho shipment)

+Mục đích: Mô tả flow nghiệp vụ (business process) phức tạp, ví dụ quy trình xử lý đơn từ tạo đến hoàn tất, các branch (giao thành công / giao thất bại / trả lại / khiếu nại).

+Nếu muốn chuyên sâu trạng thái của shipment, dùng State Machine Diagram để mô tả các trạng thái (Pending → Assigned → PickedUp → InTransit → Delivered / Failed / Returned).