**Bài 1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chức năng** | **Yêu cầu** | **Mục đích** | **Loại hệ thống** | **Giải thích** |
| Giao dịch bán hàng | Lưu đơn hàng, thanh toán, hoá đơn | Giao dịch hàng ngày | TPS | TPS xử lý các giao dịch hằng ngày: lập hoá đơn, cập nhập kho, xử lý thanh toán. |
| Phân tích xu hướng kinh doanh | Dựa trên dữ liệu để phân tích | Hỗ trợ ra quyết định | DSS | DSS giúp phân tích dữ liệu như xu hướng doanh thu, khách hàng, thị trường để hỗ trợ ra quyết định. |
| Bảng tổng quan hiệu suất tháng cho CEO | Tổng hợp doanh thu, lợi nhuận | Quản trị chiến lược | EIS | EIS cung cấp báo cáo tổng hợp cấp cao, đánh giá hiệu suất theo thời gian để quản trị chiến lược. |

**Bài 2:**

|  |  |  |
| --- | --- | --- |
| **Dự án** | **Mô hình phù hợp** | **Lý do chọn** |
| A – Phần mềm quản lý điểm cho trường cấp 2 | Waterfall (Mô hình thác nước) | - Yêu cầu **rõ ràng, ổn định**, ít thay đổi- Tính chất dự án **đơn giản**, quy trình tuyến tính phù hợp- Dễ áp dụng với **tài liệu đầy đủ** và quản lý theo từng giai đoạn |
| B – Ứng dụng mobile đặt lịch khám bệnh | Agile (Linh hoạt, Scrum) | - Yêu cầu **thay đổi thường xuyên theo phản hồi người dùng**- Có thể phát triển **từng phần (incremental)** như chức năng đăng ký, đặt lịch, thông báo...- Tối ưu thời gian phát triển, nhanh chóng ra phiên bản MVP |
| C – Hệ thống ngân hàng điện tử | Spiral (Mô hình xoắn ốc) | - Yêu cầu **độ phức tạp cao, rủi ro lớn** (bảo mật, xử lý dữ liệu nhạy cảm)- Cần đánh giá **rủi ro liên tục** trước khi phát triển thêm- Kết hợp ưu điểm của Waterfall + Prototype + Risk Analysis |

**Bài 3:**

|  |  |
| --- | --- |
| **Thành phần** | **Mô tả** |
| **Con người (People)** | Những người tham gia và vận hành hệ thống: - **Khách hàng** đặt món ăn - **Tài xế/shipper** nhận đơn, giao món - **Nhà hàng/quán ăn** quản lý menu, xác nhận đơn hàng - **Quản trị hệ thống** (admin) giám sát hoạt động, xử lý sự cố - **Nhân viên hỗ trợ (CSKH)** hỗ trợ khách hàng và đối tác |
| **Dữ liệu (Data)** | Các loại dữ liệu được hệ thống lưu trữ và xử lý: - Thông tin khách hàng (tài khoản, địa chỉ, số điện thoại) - Thông tin nhà hàng, menu, giá món ăn - Đơn hàng (món ăn đã đặt, tổng tiền, trạng thái) - Dữ liệu giao hàng (thời gian, vị trí, khoảng cách) - Thanh toán (hóa đơn, ví điện tử, lịch sử giao dịch) - Đánh giá, phản hồi của khách hàng |
| **Quy trình (Procedures)** | Các quy trình vận hành hệ thống: - Quy trình **đăng ký – đăng nhập** người dùng - Quy trình **đặt món – xác nhận – giao hàng** - Quy trình **thanh toán** (online/offline) - Quy trình **xử lý khiếu nại – hoàn tiền** - Quy trình **bảo mật dữ liệu khách hàng** |
| **Phần mềm (Software)** | Các ứng dụng và hệ thống phần mềm sử dụng: - Ứng dụng mobile cho khách hàng (Android/iOS) - Ứng dụng tài xế/đối tác giao hàng - Trang quản lý nhà hàng (restaurant dashboard) - Hệ thống backend server xử lý đơn hàng - Hệ thống thanh toán online (VNPay, MoMo, ZaloPay…) - Hệ thống bản đồ/dẫn đường (Google Maps API) |
| **Phần cứng (Hardware)** | Các thiết bị hỗ trợ hoạt động hệ thống: - Điện thoại của khách hàng và tài xế - Máy tính/desktop tại nhà hàng và trung tâm quản trị - Máy chủ (server), trung tâm dữ liệu (data center) hoặc **cloud** (AWS, GCP…) - Thiết bị mạng: WiFi, router, GPS |

**Bài 4:**

|  |  |
| --- | --- |
| **Giai đoạn** | **Việc cần làm trong dự án "Ứng dụng điểm danh"** |
| **Planning** (Lập kế hoạch) | Xác định mục tiêu hệ thống: hỗ trợ giảng viên điểm danh nhanh, lưu trữ lịch sử điểm danh, giảm gian lận. Xác định phạm vi, chi phí dự kiến, thời gian phát triển, phân chia nhân sự. |
| **Analysis** (Phân tích yêu cầu) | Thu thập yêu cầu từ giảng viên, phòng đào tạo, sinh viên. Xác định chức năng cần có: đăng nhập, quản lý lớp học, điểm danh theo buổi, xuất báo cáo. Xác định yêu cầu phi chức năng: bảo mật, hiệu năng, dễ sử dụng. |
| **Design** (Thiết kế hệ thống) | Thiết kế kiến trúc hệ thống (Client–Server), thiết kế cơ sở dữ liệu (bảng Sinh viên, Lớp học, Buổi học, Điểm danh), thiết kế giao diện ứng dụng (UI), thiết kế API nếu có. |
| **Implementation** (Lập trình/Phát triển) | Code chức năng: đăng nhập, tạo lớp học, danh sách sinh viên, điểm danh, báo cáo. Kết nối cơ sở dữ liệu. Viết API backend nếu là ứng dụng web/mobile. |
| **Testing** (Kiểm thử) | Kiểm thử chức năng (Function Test), kiểm thử giao diện, kiểm thử bảo mật đăng nhập, kiểm thử tải với nhiều người dùng, sửa lỗi trước khi triển khai. |
| **Deployment & Maintenance** (Triển khai và bảo trì) | Triển khai hệ thống lên server hoặc trường sử dụng. Đào tạo giảng viên/sinh viên sử dụng. Theo dõi hoạt động, cập nhật tính năng mới (ví dụ: điểm danh bằng QR hoặc nhận diện khuôn mặt), sửa lỗi phát sinh. |

**Bài 5:**

**1. Planning (Lập kế hoạch)**

**Mục tiêu dự án:**

* Xây dựng hệ thống điểm danh nhanh, chính xác và chống gian lận.
* Giảm thời gian điểm danh thủ công.
* Hỗ trợ theo dõi và thống kê điểm danh cho sinh viên và giảng viên.

**Phạm vi công việc:**

* Ứng dụng mobile cho sinh viên để quét QR.
* Ứng dụng web/mobile cho giảng viên tạo và quản lý buổi học.
* Hệ thống quản trị cho phòng đào tạo theo dõi thống kê.

**Nguồn lực dự kiến:**

|  |  |  |
| --- | --- | --- |
| **Vai trò** | **Số lượng** | **Nhiệm vụ** |
| PM (Quản lý dự án) | 1 | Lập kế hoạch & theo dõi dự án |
| Backend Dev | 2 | API, xử lý dữ liệu & QR |
| Frontend/Mobile Dev | 2 | App sinh viên + web giảng viên |
| Tester | 1 | Kiểm thử |
| DevOps | 1 | Triển khai |

**Rủi ro & giải pháp:**

* Gian lận điểm danh → Dùng GPS + thời gian giới hạn QR
* Truy cập nhiều gây quá tải → Sử dụng cache và scaling server
* Bảo mật dữ liệu sinh viên → Mã hóa dữ liệu & bảo mật API

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

**Use case chính:**

|  |  |
| --- | --- |
| **Actor** | **Chức năng** |
| Giảng viên | Tạo lớp học, tạo buổi học, sinh QR code, xem danh sách điểm danh |
| Sinh viên | Đăng nhập, quét QR để điểm danh |
| Phòng đào tạo (Admin) | Quản lý lớp học, thống kê điểm danh toàn trường |

**Yêu cầu chức năng (Functional Requirements)**

* Sinh viên đăng nhập bằng tài khoản trường.
* Giảng viên tạo buổi học → hệ thống sinh QR theo thời gian thực.
* Sinh viên quét mã QR trong lớp để điểm danh.
* Hệ thống ghi nhận thời gian và vị trí điểm danh.
* Admin xem báo cáo và export Excel.

**Yêu cầu phi chức năng (Non-Functional)**

* Bảo mật OAuth2 đăng nhập.
* Thời gian phản hồi < 2 giây.
* 10,000 sinh viên truy cập cùng lúc.
* Giao diện đơn giản, dễ sử dụng.
* Sao lưu dữ liệu tự động mỗi ngày.

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

1. Kiến trúc hệ thống

2. Thiết kế CSDL (mẫu)

**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 chức năng hệ thống theo góc nhìn người dùng |
| B. Mô tả lớp NguoiDung, KhoaHoc, BaiHoc và quan hệ giữa chúng | Class Diagram (Sơ đồ lớp) | Dùng để biểu diễn các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp |
| C. Mô tả luồng học viên bắt đầu → vào học → làm bài → hoàn thành | Activity Diagram (Sơ đồ hoạt động) | Dùng để mô tả quy trình xử lý hoặc luồng công việc |
| D. Mô tả cách hệ thống triển khai trên các máy chủ, thiết bị | Deployment Diagram (Sơ đồ triển khai) | Dùng để mô tả cấu trúc triển khai hệ thống trên phần cứng |
| 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 tương tác theo thời gian giữa các đối tượng |

**Bài 7:**

|  |  |
| --- | --- |
| **Giai đoạn** | **Nội dung công việc** |
| **1. Planning (Lập kế hoạch)** | - Xác định mục tiêu: hỗ trợ người dân đăng ký tiêm online, giảm tải xếp hàng trực tiếp, quản lý lịch tiêm hiệu quả.- Xác định phạm vi: đăng ký tiêm, xác nhận lịch tiêm, danh sách người đã/ chưa tiêm.- Phân tích tính khả thi (thời gian, chi phí, nhân sự).- Lập kế hoạch dự án, phân công vai trò trong nhóm. |
| **2. Requirement Analysis (Phân tích yêu cầu)** | - Thu thập yêu cầu từ trung tâm y tế.- Yêu cầu chức năng: đăng ký tài khoản, đăng ký tiêm, duyệt lịch tiêm, theo dõi trạng thái tiêm, gửi thông báo SMS/Email.- Yêu cầu phi chức năng: bảo mật thông tin cá nhân, hiệu năng cao, dễ sử dụng.- Xác định actor: Người dân, Nhân viên y tế, Admin. |
| **3. System Design (Thiết kế hệ thống)** | - Thiết kế kiến trúc phần mềm (Client – Server).- Thiết kế cơ sở dữ liệu (bảng: User, Vaccine, Registration, Schedule).- Thiết kế giao diện người dùng (UI).- Thiết kế API kết nối client-server.- Thiết kế sơ đồ UML: Use Case, Class Diagram, Sequence Diagram. |
| **4. Implementation (Lập trình/Phát triển)** | - Phát triển giao diện web/app cho người dân đăng ký tiêm.- Phát triển hệ thống quản lý cho nhân viên y tế xác nhận lịch tiêm.- Xây dựng backend: xử lý đăng ký, phân lịch, gửi thông báo.- Kết nối cơ sở dữ liệu và hoàn thiện các chức năng. |
| **5. Testing (Kiểm thử)** | - Kiểm thử chức năng (Functional testing).- Kiểm thử giao diện (UI testing).- Kiểm thử hiệu năng với nhiều người đăng ký cùng lúc.- Kiểm thử bảo mật thông tin cá nhân.- Sửa lỗi trước khi triển khai. |
| **6. Deployment & Maintenance (Triển khai và bảo trì)** | - Triển khai hệ thống lên server hoặc cloud.- Hướng dẫn người dùng (nhân viên y tế, người dân).- Hỗ trợ sử dụng và theo dõi hoạt động hệ thống.- Cập nhật tính năng (ví dụ: lịch tiêm mũi 2, quản lý chứng nhận tiêm).- Sửa lỗi phát sinh, cải tiến hiệu suất định kỳ. |

**Bài 8:**

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

|  |  |
| --- | --- |
| **Tác nhân (Actor)** | **Chức năng** |
| **Học viên** | Đăng ký tài khoản, đăng ký khóa học, học online, làm bài kiểm tra, xem điểm |
| **Giảng viên** | Quản lý khóa học, tạo bài học, đăng tài liệu, chấm điểm, theo dõi tiến độ học viên |
| **Admin** | Quản lý người dùng, phân quyền, quản lý khóa học, xem báo cáo – thống kê hệ thống |

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

|  |  |
| --- | --- |
| **Loại hệ thống thông tin** | **Vai trò trong hệ thống học trực tuyến** |
| **TPS (Transaction Processing System)** | Xử lý các giao dịch hàng ngày như đăng ký khóa học, lưu kết quả bài làm |
| **MIS (Management Information System)** | Tổng hợp báo cáo kết quả học tập và doanh thu cho quản lý trung tâm |
| **DSS (Decision Support System)** | Hỗ trợ trung tâm đưa ra quyết định như mở thêm lớp dựa trên dữ liệu đăng ký học viên |

3. **Mô hình phát triển**

**Chọn mô hình: Agile (Scrum)**

**Lý do:**

* Yêu cầu hệ thống có thể **thay đổi thường xuyên** (thêm bài học, thêm tính năng quiz, livestream,...).
* Có thể phát triển **theo từng sprint** và ra **phiên bản MVP sớm**.
* Dễ dàng **thu thập phản hồi từ người dùng** (giảng viên/học viên) để cải tiến liên tục.
* Phù hợp với **dự án giáo dục trực tuyến hiện đại**, ưu tiên cập nhật nhanh.

**4. Nêu 3 sơ đồ UML sẽ sử dụng khi thiết kế hệ thống**

|  |  |
| --- | --- |
| **UML Diagram** | **Mục đích sử dụng** |
| **Use Case Diagram** | Xác định chức năng hệ thống theo hành vi của từng tác nhân |
| **Class Diagram** | Thiết kế cấu trúc dữ liệu (User, Course, Lesson, Enrollment, Score...) |
| **Sequence Diagram** | Mô tả luồng tương tác trong hệ thống, ví dụ: học viên đăng ký khóa học |

**Bài 9:**

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

|  |  |
| --- | --- |
| **Actors (Tác nhân)** | **Chức năng chính** |
| **Khách hàng (Customer)** | - Tạo đơn hàng (nhập thông tin người gửi/nhận, địa chỉ, dịch vụ)- Thanh toán phí giao hàng (nếu có)- Theo dõi trạng thái đơn hàng theo thời gian thực- Yêu cầu hỗ trợ: khiếu nại, hủy đơn, thay đổi thông tin- Nhận thông báo qua SMS/Email/App |
| **Nhân viên vận chuyển (Courier/Driver)** | - Nhận đơn giao từ hệ thống (assign job)- Xác nhận lấy hàng (pickup)- Cập nhật trạng thái vận chuyển- Ghi nhận chứng từ giao hàng (ảnh, chữ ký điện tử)- Gửi vị trí GPS thường xuyên |
| **Nhân viên kho / Điều phối (Warehouse / Dispatcher)** | - Tiếp nhận hàng tại kho trung tâm- Phân loại và gom tuyến giao hàng- Điều phối tài xế và tuyến đường- Quản lý chuyển tuyến (hub → hub hoặc hub → customer)- Kiểm soát hàng tồn, hàng trả lại |
| **Quản lý vận hành (Operations Manager)** | - Theo dõi KPI giao hàng- Giám sát hiệu suất nhân viên và tuyến giao- Xử lý sự cố giao hàng (tắc tuyến, đơn trễ)- Quản lý chất lượng dịch vụ- Xem báo cáo vận hành chi tiết |
| **CEO / Ban điều hành (Executive)** | - Xem dashboard tổng quan theo thời gian thực- Theo dõi doanh thu – chi phí – lợi nhuận- Đánh giá vùng tăng trưởng – vùng quá tải- Xem xu hướng nhu cầu dịch vụ logistics- Ra quyết định chiến lược |
| **Hệ thống bên thứ 3 (Third-party Systems)** | - Xử lý thanh toán (VNPay, Momo, Stripe...)- Cung cấp bản đồ và GPS (Google Maps API)- Tính toán thời gian dự kiến (ETA)- Gửi thông báo SMS/Email- Tích hợp API hãng vận chuyển khác (nếu có) |

**2. Phân loại từng chức năng theo hệ thống thông tin**

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Loại hệ thống phù hợp** | **Lý do ngắn** |
| Tạo đơn hàng, thanh toán, xác nhận pickup | **TPS** | Giao dịch hàng ngày, cần luôn nhất quán và siêu tin cậy |
| Cập nhật trạng thái giao hàng (driver) & tracking GPS | **TPS** | Ghi nhận trạng thái theo thời gian thực — transaction processing |
| Ghi nhận bằng chứng giao hàng (ảnh, chữ ký) | **TPS** | Lưu bằng chứng giao, liên quan đến giao dịch |
| Theo dõi đơn của khách hàng (portal/notifications) | **TPS / MIS** | TPS cho dữ liệu thời gian thực; MIS nếu tổng hợp lịch sử cho báo cáo |
| Báo cáo hiệu suất vận hành hàng ngày/tuần (KPI) | **MIS** | Tổng hợp và trình bày cho quản lý vận hành |
| Phân tích vùng tồn đọng, vùng tắc nghẽn (heatmap) | **DSS** | Phân tích dữ liệu để hỗ trợ quyết định tối ưu phân bổ nguồn lực |
| Dự báo nhu cầu theo vùng / tối ưu phân tuyến | **DSS** | Mô phỏng & phân tích để hỗ trợ lập kế hoạch (what-if) |
| Dashboard tổng quan cấp CEO: số đơn, vùng hoạt động, tỉ lệ giao đúng hẹn | **EIS** | Cung cấp tóm tắt chiến lược, trực quan cho lãnh đạo |
| Báo cáo tài chính / doanh thu tổng hợp | **MIS / EIS** | MIS cho quản lý, EIS cho lãnh đạo khi cần tầm nhìn chiến lược |
| Hệ thống cảnh báo rủi ro (tăng đột biến khiếu nại) | **DSS / MIS** | DSS nếu cần phân tích nguyên nhân; MIS nếu là cảnh báo định kỳ |

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

**Đề xuất mô hình: Agile (Scrum) kết hợp DevOps** — *lý do:*

* **Tính thay đổi cao:** nghiệp vụ logistics thường thay đổi (tích hợp đối tác mới, thay đổi luồng vận hành, quy tắc định giá). Agile cho phép phản hồi nhanh và lặp cải tiến.
* **Phát hành incremental / MVP:** có thể nhanh ra bản MVP xử lý tạo đơn & tracking, sau đó thêm tối ưu tuyến, BI, báo cáo.
* **Tương tác liên tục với stakeholders:** dispatcher, drivers, khách hàng, quản lý cần được thu feedback sớm — phù hợp Scrum.
* **Yêu cầu vận hành liên tục & ổn định:** tích hợp **DevOps** (CI/CD, monitoring, autoscaling) đảm bảo triển khai nhanh, rollback an toàn, và theo dõi hệ thống (critical cho TPS).
* **Khả năng mở rộng:** kiến trúc microservices + Agile giúp tách module (order, tracking, billing, reporting) độc lập phát triển và scale.

**4. Đề xuất 4 sơ đồ UML sẽ sử dụng + mục đích ngắn**

**a Use Case Diagram**

**Mục đích:** Mô tả các actor và chức năng chính hệ thống (tạo đơn, tracking, cập nhật trạng thái, báo cáo). Dùng để thảo luận nhanh với stakeholders và xác nhận phạm vi.

**b Class Diagram**

**Mục đích:** Thiết kế cấu trúc dữ liệu chính (Order, Customer, Shipment, Route, Driver, Warehouse, ProofOfDelivery, Payment, Notification). Thể hiện thuộc tính, phương thức, và quan hệ (aggregation/composition, associations).

**c Sequence Diagram (ví dụ: khi khách hàng tạo đơn & hệ thống phân tuyến)**

**Mục đích:** Mô tả luồng tương tác theo thời gian giữa các thành phần: Customer App → Order Service → Routing Engine → Dispatcher → Driver App → Notification Service. Rõ ràng cho developer hiểu order of operations, API calls, trả về lỗi.

**d Deployment Diagram (hoặc Activity Diagram nếu muốn biểu diễn workflow chi tiết)**

* Mình khuyến nghị **Deployment Diagram** nếu mục tiêu là thiết kế hạ tầng: thể hiện server, load balancer, database cluster, message queue, services bên thứ 3 (GPS, payment).
* **Mục đích:** Giúp đội DevOps/Infra hiểu cách triển khai (microservices trên k8s, DB cluster, Redis cache, message broker, CDN…).
* (Nếu bạn ưu tiên mô tả quy trình nghiệp vụ thì thay bằng **Activity Diagram** để mô tả luồng nghiệp vụ như xử lý khiếu nại / xử lý giao thất bại.)