EX1:

**1. Chức năng: Giao dịch bán hàng**

**Loại hệ thống tương ứng: Hệ thống Xử lý Giao dịch (TPS - Transaction Processing System)**

* TPS được thiết kế để thu thập, xử lý, lưu trữ và truy xuất dữ liệu từ các giao dịch hàng ngày một cách nhanh chóng và chính xác.
* Ví dụ điển hình là các hệ thống máy bán hàng (POS), hệ thống đặt hàng online, hoặc phần mềm ghi nhận hóa đơn. Mục tiêu chính của TPS là đảm bảo tính toàn vẹn của dữ liệu giao dịch. Dữ liệu từ TPS chính là nguồn đầu vào quan trọng cho các hệ thống cấp cao hơn.

**2. Chức năng: Phân tích xu hướng kinh doanh**

**Loại hệ thống tương ứng: Hệ thống Hỗ trợ ra Quyết định (DSS - Decision Support System)**

* DSS sử dụng dữ liệu (thường là từ TPS) kết hợp với các mô hình phân tích, công cụ thống kê để giúp các nhà quản lý cấp trung đưa ra các quyết định bán cấu trúc (semi-structured decisions).
* Hệ thống này cho phép bạn thực hiện các truy vấn phức tạp, phân tích "what-if" (nếu... thì sao...), và khám phá dữ liệu từ nhiều góc độ khác nhau để tìm ra insight, thay vì chỉ xem các báo cáo có sẵn

**3. Chức năng: Bảng tổng quan hiệu suất hàng tháng cho CEO**

**Loại hệ thống tương ứng: Hệ thống Thông tin cho Lãnh đạo (EIS - Executive Information System)**

* EIS là một dạng đặc biệt của DSS, được thiết kế riêng cho các nhà lãnh đạo cấp cao.
* Nó tập trung vào việc trình bày thông tin một cách cực kỳ trực quan (qua biểu đồ, đồ thị), dễ theo dõi và cho phép xem chi tiết (drill-down) khi cần.
* Ví dụ, từ biểu đồ tổng doanh thu tháng trên dashboard, CEO có thể nhấp vào để xem doanh thu theo từng khu vực hoặc dòng sản phẩm. Mục tiêu của EIS là cung cấp thông tin chiến lược, giúp lãnh đạo nắm bắt tình hình nhanh chóng mà không cần đi sâu vào chi tiết kỹ thuật.

EX2

**Dự án A: Phần mềm quản lý điểm cho trường cấp 2**

* **Đặc điểm:** Yêu cầu rõ ràng, ổn định. Nghiệp vụ quản lý điểm đã được chuẩn hóa và ít có khả năng thay đổi trong quá trình phát triển.
* **Lựa chọn: Mô hình Thác nước (Waterfall)**
* **Lý do:**
  + Với các yêu cầu đã được xác định rõ từ đầu, việc áp dụng quy trình tuần tự của Waterfall sẽ giúp dự án được quản lý một cách chặt chẽ, dễ dàng theo dõi tiến độ và đảm bảo chất lượng theo đúng kế hoạch.
  + Không cần sự linh hoạt cao.

**Dự án B: Ứng dụng mobile đặt lịch khám bệnh**

* **Đặc điểm:** Yêu cầu cần sự linh hoạt cao. Thị trường ứng dụng di động thay đổi nhanh, cần liên tục cập nhật để đáp ứng phản hồi của người dùng (bệnh nhân, bác sĩ) và cạnh tranh.
* **Lựa chọn: Mô hình Linh hoạt (Agile)**
* **Lý do:**
  + Agile cho phép phát triển và phát hành nhanh một phiên bản cơ bản (MVP - Minimum Viable Product) để thu thập phản hồi sớm.
  + Các yêu cầu mới hoặc thay đổi (ví dụ: thêm tính năng tư vấn online, thanh toán viện phí,...) có thể dễ dàng được đưa vào các chu kỳ phát triển tiếp theo mà không làm ảnh hưởng lớn đến toàn bộ dự án.

**Dự án C: Hệ thống ngân hàng điện tử**

* **Đặc điểm:** Độ phức tạp và yêu cầu bảo mật cực kỳ cao. Bất kỳ sai sót nào cũng có thể gây ra thiệt hại tài chính nghiêm trọng. Đây là một dự án lớn và tiềm ẩn nhiều rủi ro.
* **Lựa chọn: Mô hình Xoắn ốc (Spiral)**
* **Lý do:**
  + Điểm mạnh nhất của mô hình Xoắn ốc là việc đặt "phân tích rủi ro" làm trọng tâm trong mỗi vòng lặp. Điều này cực kỳ quan trọng đối với một hệ thống tài chính, nơi các rủi ro về bảo mật, hiệu năng, và tính chính xác của giao dịch phải được xác định và xử lý sớm.
  + Việc xây dựng các phiên bản nguyên mẫu (prototype) ở mỗi vòng lặp giúp kiểm tra và xác nhận các chức năng phức tạp trước khi xây dựng hoàn chỉnh.

EX 3:

|  |  |
| --- | --- |
| Con Người | Là những đối tượng tương tác và vận hành hệ thống. Mỗi nhóm có vai trò và nhu cầu riêng biệt. Bao gồm: - **Khách hàng:** Người dùng cuối, duyệt menu, đặt món, thanh toán và đánh giá. - **Đối tác nhà hàng/quán ăn:** Nhân viên và quản lý nhà hàng, tiếp nhận và chuẩn bị đơn hàng. - **Tài xế (Shipper):** Người nhận và giao đơn hàng từ nhà hàng đến khách hàng. - **Quản trị viên hệ thống:** Đội ngũ kỹ sư, nhân viên hỗ trợ, quản lý vận hành, đảm bảo hệ thống hoạt động trơn tru. |
| Dữ Liệu | Là toàn bộ thông tin được tạo ra, xử lý, và lưu trữ trong hệ thống. Dữ liệu là tài sản cốt lõi, giúp cá nhân hóa trải nghiệm và tối ưu vận hành. Bao gồm: - **Dữ liệu người dùng:** Thông tin tài khoản, lịch sử đặt hàng, địa chỉ, phương thức thanh toán. - **Dữ liệu nhà hàng:** Thực đơn (tên món, giá, hình ảnh), địa chỉ, giờ hoạt động, đánh giá. - **Dữ liệu đơn hàng:** Chi tiết món ăn, tổng tiền, trạng thái đơn hàng (đang xử lý, đang giao, đã giao). - **Dữ liệu vị trí:** Dữ liệu GPS theo thời gian thực của tài xế và khách hàng để theo dõi quá trình giao hàng. |
| Quy Trình | Là luồng các bước và nghiệp vụ được thực hiện trên hệ thống để hoàn thành một mục tiêu cụ thể (ví dụ: hoàn thành một đơn hàng). Quy trình chuẩn hóa giúp hệ thống hoạt động hiệu quả. Quy trình đặt món cơ bản gồm: 1. **Tìm kiếm & Chọn món:** Khách hàng tìm nhà hàng, duyệt menu và thêm món vào giỏ hàng. 2. **Đặt hàng & Thanh toán:** Khách hàng xác nhận đơn hàng và chọn phương thức thanh toán. 3. **Xử lý đơn hàng:** Hệ thống gửi thông báo đến nhà hàng. Nhà hàng xác nhận và chuẩn bị món. 4. **Giao hàng:** Hệ thống tìm và gán tài xế. Tài xế đến nhà hàng lấy món và giao cho khách. 5. **Hoàn tất & Đánh giá:** Khách hàng nhận món, đơn hàng được cập nhật hoàn tất. Khách hàng có thể để lại đánh giá. |
| Phần Mềm | Là các ứng dụng và chương trình máy tính tạo nên giao diện và xử lý logic cho hệ thống. - **Ứng dụng cho khách hàng (Client App):** Giao diện trên điện thoại để người dùng đặt món. - **Ứng dụng cho nhà hàng (Merchant App):** Ứng dụng (thường trên máy tính bảng) để nhà hàng nhận và quản lý đơn. - **Ứng dụng cho tài xế (Driver App):** Ứng dụng trên điện thoại để tài xế nhận cuốc xe, xem bản đồ và quản lý thu nhập. - **Hệ thống lõi (Backend):** Máy chủ xử lý logic, quản lý cơ sở dữ liệu, thuật toán điều phối tài xế, cổng thanh toán. |
| Phần Cứng | Là các thiết bị vật lý cần thiết để hệ thống có thể hoạt động. - **Điện thoại thông minh (Smartphone):** Của khách hàng và tài xế để cài đặt và sử dụng ứng dụng. - **Máy tính bảng/POS:** Tại các nhà hàng để nhận đơn hàng. - **Hệ thống máy chủ (Servers):** Các máy chủ mạnh mẽ để lưu trữ dữ liệu và chạy phần mềm backend. - **Thiết bị mạng:** Router, switch, và các hạ tầng mạng khác để kết nối tất cả các thành phần. |

Ex 4:

|  |  |
| --- | --- |
| Giai Đoạn | Việc cần Làm trong dự Án |
| Planning | - **Xác định mục tiêu:** Xây dựng ứng dụng giúp điểm danh nhanh chóng, chính xác, và dễ dàng xuất báo cáo. - **Phạm vi dự án:** Ứng dụng sẽ bao gồm các chức năng cho giảng viên (tạo lớp, điểm danh, xem lịch sử) và sinh viên (xem trạng thái điểm danh). - **Lập kế hoạch:** Ước tính chi phí, phân bổ nhân sự, và xác định các mốc thời gian quan trọng. |
| Analysis | - **Thu thập yêu cầu:** Phỏng vấn giảng viên, sinh viên và phòng đào tạo để hiểu rõ nhu cầu. Ví dụ: Cần điểm danh bằng QR code hay thủ công? Báo cáo cần những thông tin gì? - **Phân tích yêu cầu:** Mô hình hóa các chức năng (use case), xác định các quy tắc nghiệp vụ (ví dụ: sinh viên được phép điểm danh trong bao lâu?). - **Tài liệu hóa:** Viết tài liệu đặc tả yêu cầu phần mềm (SRS - Software Requirement Specification). |
| Design | - **Thiết kế kiến trúc:** Lựa chọn công nghệ (ví dụ: React Native cho mobile, Node.js cho backend). Quyết định kiến trúc tổng thể (ví dụ: Microservices). - **Thiết kế cơ sở dữ liệu:** Tạo sơ đồ quan hệ thực thể (ERD) cho các bảng như Students, Classes, AttendanceRecords. - **Thiết kế giao diện (UI/UX):** Tạo wireframe và mockup cho các màn hình của ứng dụng, đảm bảo dễ sử dụng cho cả giảng viên và sinh viên. |
| Implementation | - **Phát triển Backend:** Xây dựng cơ sở dữ liệu, viết các API để xử lý logic (ví dụ: API tạo QR code, API lưu thông tin điểm danh). - **Phát triển Frontend:** Lập trình giao diện ứng dụng di động dựa trên thiết kế UI/UX đã duyệt. - **Tích hợp:** Kết nối ứng dụng di động với backend thông qua các API. |
| Testing | - **Unit Test:** Lập trình viên tự kiểm tra từng hàm, từng module nhỏ mình viết. - **Integration Test:** Kiểm tra sự tương tác giữa frontend và backend có hoạt động đúng không. - **System Test:** Kiểm thử toàn bộ hệ thống để đảm bảo tất cả các chức năng hoạt động như mong đợi. - **User Acceptance Test (UAT):** Mời một nhóm giảng viên và sinh viên dùng thử và cho phản hồi. |
| Deployment & Maintenance | - **Triển khai (Deployment):** Đưa ứng dụng lên các kho ứng dụng (Google Play Store, Apple App Store). Cài đặt backend lên máy chủ. - **Đào tạo & Hỗ trợ:** Hướng dẫn giảng viên và sinh viên cách sử dụng. Cung cấp kênh hỗ trợ khi có sự cố. - **Bảo trì (Maintenance):** Theo dõi hệ thống, sửa lỗi phát sinh, và lên kế hoạch cập nhật các phiên bản mới với các tính năng cải tiến. |

Ex 5

**Giai đoạn 1: Planning (Hoạch định)**

Ở giai đoạn khởi đầu này, mục tiêu là xác định tính khả thi và vạch ra lộ trình tổng thể cho dự án.

* **Xác định mục tiêu & Vấn đề:**
  + **Mục tiêu:** Xây dựng một hệ thống điểm danh tự động, chính xác và hiệu quả bằng công nghệ QR code, thay thế cho phương pháp thủ công tốn thời gian và dễ sai sót.
  + **Vấn đề giải quyết:** Giảm thời gian điểm danh đầu giờ, loại bỏ tình trạng điểm danh hộ, cung cấp số liệu thống kê tức thì cho phòng đào tạo.
* **Xác định phạm vi:**
  + **Trong phạm vi (In-scope):**
    - **Giảng viên:** Đăng nhập, tạo buổi học, tạo mã QR code duy nhất cho mỗi buổi, xem danh sách sinh viên đã điểm danh.
    - **Sinh viên:** Đăng nhập, dùng camera của ứng dụng để quét mã QR, xem lại lịch sử điểm danh của bản thân.
    - **Phòng đào tạo:** Đăng nhập vào một trang quản trị (web portal), xem và xuất báo cáo thống kê điểm danh theo lớp, theo giảng viên, hoặc theo sinh viên.
  + **Ngoài phạm vi (Out-of-scope):** Các chức năng quản lý học phí, đăng ký môn học, hay quản lý lịch thi.
* **Lập kế hoạch sơ bộ:**
  + **Nhân sự:** 1 Project Manager, 2 Backend Devs, 1 Mobile Dev (React Native), 1 Web Dev (cho portal), 1 QC/Tester.
  + **Thời gian:** Ước tính 3 tháng để hoàn thành phiên bản đầu tiên.
  + **Chi phí:** Ước tính chi phí phát triển, chi phí máy chủ và chi phí bảo trì.

**Giai đoạn 2: Requirement Analysis (Phân tích Yêu cầu)**

Giai đoạn này tập trung vào việc thu thập và làm rõ các yêu cầu chi tiết từ các bên liên quan.

* **Thu thập yêu cầu:**
  + Tổ chức các buổi họp với phòng đào tạo, giảng viên và nhóm sinh viên để lấy ý kiến.
  + Khảo sát về các quy trình điểm danh hiện tại và các khó khăn gặp phải.
  + Xác định các yêu cầu chức năng (sinh viên phải ở trong phạm vi lớp học mới điểm danh được) và phi chức năng (hệ thống phải xử lý đồng thời 500 sinh viên điểm danh trong 1 phút).
* **Phân tích và Mô hình hóa yêu cầu:**
  + Viết tài liệu đặc tả yêu cầu chi tiết (SRS).
  + **Sử dụng Sơ đồ UML:**
    - **Sơ đồ Use Case (Use Case Diagram):** Đây là sơ đồ quan trọng nhất ở giai đoạn này. Nó sẽ mô tả trực quan các hành động (use case) mà mỗi tác nhân (actor: Giảng viên, Sinh viên, Phòng đào tạo) có thể thực hiện trên hệ thống.
      * *Ví dụ: Actor "Sinh viên" có các use case như "Đăng nhập", "Quét mã QR", "Xem lịch sử điểm danh".*
    - **Sơ đồ Hoạt động (Activity Diagram):** Dùng để mô tả chi tiết luồng công việc cho một Use Case phức tạp.
      * *Ví dụ: Mô tả luồng "Điểm danh bằng QR", bắt đầu từ khi giảng viên tạo mã, sinh viên quét mã, hệ thống xác thực và lưu kết quả.*

**Giai đoạn 3: System Design (Thiết kế Hệ thống)**

Dựa trên các yêu cầu đã phân tích, giai đoạn này sẽ chuyển hóa chúng thành một bản thiết kế chi tiết cho hệ thống.

* **Thiết kế kiến trúc (Architectural Design):**
  + Quyết định kiến trúc tổng thể: Client-Server.
  + **Client:** Ứng dụng di động cho sinh viên và giảng viên; Một trang web (Web Portal) cho phòng đào tạo.
  + **Server (Backend):** Một hệ thống API để xử lý logic, xác thực và tương tác với cơ sở dữ liệu.
  + **Cơ sở dữ liệu:** Lựa chọn hệ quản trị CSDL (ví dụ: PostgreSQL, MySQL).
* **Thiết kế chi tiết (Detailed Design):**
  + **Sử dụng Sơ đồ UML:**
    - **Sơ đồ Lớp (Class Diagram):** Dùng để thiết kế cấu trúc tĩnh của hệ thống. Nó định nghĩa các lớp (Classes) chính, các thuộc tính (attributes) và mối quan hệ giữa chúng.
      * *Ví dụ: Các lớp như Student, Lecturer, Course, AttendanceSession, AttendanceRecord và mối quan hệ giữa chúng (một Course có nhiều AttendanceSession).*
    - **Sơ đồ Tuần tự (Sequence Diagram):** Mô tả sự tương tác giữa các đối tượng theo một trật tự thời gian để thực hiện một chức năng cụ thể.
      * *Ví dụ: Sơ đồ cho thấy các bước từ khi sinh viên nhấn nút "Quét QR" trên app, app gửi yêu cầu đến API, API xác thực mã QR, sau đó ghi vào CSDL và trả kết quả về cho app.*
* **Thiết kế Giao diện (UI/UX Design):**
  + Tạo wireframe và mockup chi tiết cho từng màn hình của ứng dụng di động và trang web quản trị.

Ex 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** | **Sơ đồ Use Case (Use Case Diagram)** *Sơ đồ này tập trung vào việc mô tả các chức năng mà hệ thống cung cấp cho người dùng (tác nhân).* |
| **B. Mô tả lớp NguoiDung, KhoaHoc, BaiHoc và quan hệ giữa chúng** | **Sơ đồ Lớp (Class Diagram)** *Sơ đồ này dùng để mô tả cấu trúc tĩnh của hệ thống, bao gồm các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng.* |
| **C. Mô tả luồng học viên bắt đầu → vào học → làm bài → hoàn thành** | **Sơ đồ Hoạt động (Activity Diagram)** *Sơ đồ này rất phù hợp để mô tả các luồng công việc (workflow), quy trình nghiệp vụ hoặc các bước tuần tự của một hành động.* |
| **D. Mô tả cách hệ thống triển khai trên các máy chủ, thiết bị** | **Sơ đồ Triển khai (Deployment Diagram)** *Sơ đồ này mô tả kiến trúc vật lý của hệ thống, cho thấy các thành phần phần mềm được triển khai trên các nút phần cứng (máy chủ, thiết bị) như thế nào.* |
| **E. Mô tả thứ tự tương tác giữa học viên và hệ thống khi nộp bài** | **Sơ đồ Tuần tự (Sequence Diagram)** *Sơ đồ này nhấn mạnh vào thứ tự thời gian của các thông điệp được trao đổi giữa các đối tượng để thực hiện một chức năng cụ thể.* |

Ex 7

|  |  |
| --- | --- |
| **Giai đoạn** | **Kế hoạch thực hiện cho dự án "Đăng ký tiêm chủng online"** |
| **1. Planning (Hoạch định)** | - **Xác định mục tiêu:** Số hóa quy trình đăng ký, giảm tải công việc giấy tờ, cung cấp dữ liệu chính xác và kịp thời về tình hình tiêm chủng. - **Phạm vi dự án:**    + **Người dân:** Đăng ký thông tin cá nhân, chọn loại vắc-xin (nếu có), gửi yêu cầu hẹn lịch.    + **Trung tâm y tế:** Tiếp nhận, xác thực đơn đăng ký, xếp lịch hẹn và gửi thông báo.    + **Quản lý:** Theo dõi, thống kê và xuất báo cáo danh sách đã tiêm/chưa tiêm. - **Kế hoạch nguồn lực:** Xác định đội ngũ (Quản lý dự án, Lập trình viên, Kỹ sư kiểm thử), ước tính ngân sách và xây dựng một lộ trình thời gian sơ bộ cho từng giai đoạn. |
| **2. Analysis (Phân tích Yêu cầu)** | - **Thu thập yêu cầu nghiệp vụ:** Tổ chức các buổi làm việc với đội ngũ y bác sĩ, nhân viên hành chính của trung tâm y tế và khảo sát một nhóm người dân để hiểu rõ quy trình và nhu cầu thực tế. - **Phân tích yêu cầu chức năng:**    + Người dân có thể sửa thông tin đăng ký trước khi được xác nhận.    + Hệ thống tự động gửi SMS/Email nhắc lịch tiêm.    + Quản lý có thể lọc danh sách theo độ tuổi, loại vắc-xin, trạng thái tiêm. - **Phân tích yêu cầu phi chức năng:** Hệ thống phải đảm bảo **bảo mật tuyệt đối** thông tin y tế cá nhân, hoạt động ổn định và dễ dàng sử dụng cho cả người không rành công nghệ. - **Vẽ Sơ đồ Use Case:** Mô tả các tác nhân (Người dân, Nhân viên y tế, Quản lý) và các hành động họ có thể thực hiện. |
| **3. Design (Thiết kế Hệ thống)** | - **Thiết kế kiến trúc:** Lựa chọn mô hình Client-Server. Xây dựng một trang web (Web App) đáp ứng cho cả máy tính và di động. - **Thiết kế Cơ sở dữ liệu (Database Design):** Thiết kế các bảng dữ liệu chính như Citizens (Thông tin người dân), Vaccinations (Lịch sử tiêm), Appointments (Lịch hẹn). Sử dụng **Sơ đồ Lớp (Class Diagram)** để mô hình hóa. - **Thiết kế Giao diện (UI/UX):** Tạo wireframe và mockup cho các luồng chính: luồng đăng ký của người dân, luồng duyệt đơn và xếp lịch của nhân viên y tế. Ưu tiên sự đơn giản và rõ ràng. - **Vẽ Sơ đồ Tuần tự (Sequence Diagram):** Mô tả chi tiết các bước tương tác để hoàn thành một chức năng, ví dụ: luồng "Xác nhận lịch hẹn". |
| **4. Implementation (Thi công)** | - **Lập trình Backend:** Xây dựng logic nghiệp vụ, các API (ví dụ: API nhận đơn đăng ký, API gửi thông báo), và kết nối với cơ sở dữ liệu. - **Lập trình Frontend:** Phát triển giao diện người dùng dựa trên thiết kế UI/UX đã được duyệt, đảm bảo tính tương tác và thân thiện. - **Tích hợp hệ thống:** Kết nối các thành phần Frontend và Backend lại với nhau. Tích hợp với các dịch vụ bên thứ ba nếu cần (ví dụ: dịch vụ gửi SMS). |
| **5. Testing (Kiểm thử)** | - **Kiểm thử chức năng:** Đảm bảo tất cả các chức năng từ đăng ký, xác nhận, đến xuất báo cáo đều hoạt động đúng như đặc tả. - **Kiểm thử bảo mật:** Rà soát các lỗ hổng có thể làm rò rỉ thông tin nhạy cảm của người dân. - **Kiểm thử hiệu năng:** Giả lập tình huống có hàng ngàn người truy cập đăng ký cùng lúc để đảm bảo hệ thống không bị quá tải. - **Thử nghiệm người dùng (UAT):** Mời một nhóm nhân viên y tế và người dân dùng thử hệ thống trên môi trường giả lập để thu thập phản hồi cuối cùng. |
| **6. Deployment & Maintenance (Triển khai & Bảo trì)** | - **Triển khai:** Cài đặt hệ thống lên máy chủ chính thức và cho phép người dân bắt đầu truy cập, đăng ký. - **Đào tạo & Chuyển giao:** Tổ chức các buổi hướng dẫn sử dụng chi tiết cho đội ngũ nhân viên của trung tâm y tế. - **Giám sát & Hỗ trợ:** Theo dõi hoạt động của hệ thống 24/7 để phát hiện và xử lý sự cố kịp thời. Thiết lập kênh hỗ trợ người dùng. - **Bảo trì & Nâng cấp:** Sửa các lỗi phát sinh sau khi triển khai và lên kế hoạch cho các phiên bản cập nhật trong tương lai dựa trên phản hồi thực tế. |

Ex 8:

**1. Xác định Tác nhân (Actors) và Chức năng**

Hệ thống có ba tác nhân chính, mỗi tác nhân có những vai trò và chức năng riêng biệt:

|  |  |
| --- | --- |
| **Tác nhân (Actor)** | **Chức năng chính** |
| **Học viên** | - Đăng ký/Đăng nhập vào hệ thống. - Xem danh sách các khóa học. - Đăng ký tham gia khóa học. - Truy cập tài liệu, video bài giảng. - Nộp bài tập và làm bài kiểm tra. - Xem điểm số và tiến độ học tập. |
| **Giảng viên** | - Đăng nhập vào hệ thống. - Tạo và quản lý nội dung khóa học (bài giảng, bài tập). - Chấm điểm và đưa ra phản hồi cho bài tập của học viên. - Theo dõi tiến độ học tập của từng học viên trong lớp mình phụ trách. - Giao tiếp với học viên (qua diễn đàn, tin nhắn). |
| **Admin (Quản trị viên)** | - Quản lý tài khoản người dùng (tạo, xóa, sửa). - Phân quyền chi tiết cho Giảng viên và Học viên. - Quản lý danh mục khóa học. - Theo dõi và xem các báo cáo, thống kê (số lượng học viên, doanh thu...). - Cấu hình các thiết lập chung của hệ thống. |

**2. Phân loại Hệ thống Thông tin**

Dựa trên các chức năng, hệ thống này là sự kết hợp của nhiều loại hệ thống thông tin:

* **Hệ thống Xử lý Giao dịch (TPS - Transaction Processing System):** Nền tảng của hệ thống, xử lý các giao dịch hàng ngày như đăng ký khóa học của học viên, giảng viên nhập điểm, admin tạo tài khoản. Các hoạt động này cần được ghi nhận một cách chính xác và nhanh chóng.
* **Hệ thống Thông tin Quản lý (MIS - Management Information System):** Lớp cao hơn TPS, hệ thống này tổng hợp dữ liệu từ các giao dịch để tạo ra các báo cáo định kỳ cho Giảng viên và Admin. Ví dụ: Báo cáo tỷ lệ hoàn thành khóa học, báo cáo điểm số trung bình của một lớp, thống kê số lượng học viên mới hàng tháng.
* **Hệ thống Hỗ trợ Ra quyết định (DSS - Decision Support System):** Hệ thống có thể được phát triển thêm các tính năng DSS để giúp Admin đưa ra quyết định chiến lược, ví dụ như phân tích dữ liệu để tìm ra "khóa học nào đang được quan tâm nhất" hoặc "giảng viên nào có tỷ lệ học viên hài lòng cao nhất" để có kế hoạch kinh doanh phù hợp.

**3. Đề xuất Mô hình Phát triển Phần mềm**

Mô hình phù hợp nhất cho dự án này là **Mô hình Agile (Linh hoạt)**, cụ thể là Scrum.

**Lý do lựa chọn:**

1. **Yêu cầu dễ thay đổi:** Lĩnh vực giáo dục trực tuyến luôn thay đổi. Agile cho phép dễ dàng điều chỉnh và bổ sung các tính năng mới (ví dụ: thêm lớp học livestream, gamification) để đáp ứng nhu cầu của học viên và thị trường.
2. **Cung cấp giá trị sớm:** Thay vì chờ đợi hàng tháng để có sản phẩm hoàn chỉnh, Agile cho phép ra mắt một phiên bản tối thiểu (MVP) với các chức năng cốt lõi (đăng ký, xem bài học). Sau đó, các tính năng mới sẽ được bổ sung và hoàn thiện qua từng chu kỳ phát triển ngắn (sprint).
3. **Lấy phản hồi liên tục:** Mô hình này cho phép đội ngũ phát triển nhận phản hồi thường xuyên từ chính các học viên, giảng viên và admin, đảm bảo sản phẩm cuối cùng thực sự hữu ích và thân thiện với người dùng.

**4. Ba Sơ đồ UML chính sẽ sử dụng**

Khi đi sâu vào thiết kế, ba loại sơ đồ UML sau đây sẽ là công cụ không thể thiếu:

1. **Sơ đồ Use Case (Use Case Diagram):**
   * **Mục đích:** Mô tả trực quan "ai làm gì" trên hệ thống. Sơ đồ này sẽ được tạo ra đầu tiên để xác nhận lại các chức năng chính đã được xác định ở mục 1 và đảm bảo tất cả các bên liên quan có chung một cách hiểu về phạm vi của dự án.
2. **Sơ đồ Lớp (Class Diagram):**
   * **Mục đích:** Mô tả cấu trúc dữ liệu của hệ thống. Sơ đồ này sẽ định nghĩa các lớp đối tượng chính như HocVien, GiangVien, KhoaHoc, BaiHoc, DiemSo cùng với các thuộc tính và mối quan hệ giữa chúng. Đây là nền tảng cho việc thiết kế cơ sở dữ liệu.
3. **Sơ đồ Tuần tự (Sequence Diagram):**
   * **Mục đích:** Mô tả chi tiết sự tương tác giữa các đối tượng theo thứ tự thời gian để thực hiện một chức năng cụ thể. Ví dụ, sơ đồ này sẽ minh họa các bước xảy ra khi một học viên "Nộp bài tập": từ giao diện người dùng, qua lớp xử lý logic, đến việc lưu trữ vào cơ sở dữ liệu và trả về thông báo thành công.