**SESSION 01**

**Bài 1:**

*\*Giao dịch bán hàng – TPS (Transaction Processing System)*

=> Giải thích**:**

* Giao dịch bán hàng là hoạt động giao tiếp trực tiếp với khách hàng với các giao dịch lặp (bán hàng, trả hàng, thanh toán, xuất hoá đơn...).
* TPS xử lý khối lượng giao dịch lớn, theo (gần) thời gian thực, đảm bảo tính chính xác, nhất quán
* TPS cung cấp các tác nghiệp cơ bản hàng ngày, có tỉnh tự động hóa: nhập liệu đơn

*\* Phân tích xu hướng kinh doanh — DSS (Decision Support System)*=> Giải thích:

* Phân tích chuyên sâu xu hướng cần xử lý, tổng hợp dữ liệu lịch sử (từ TPS, CRM, kho, marketing)
* Dùng mô hình toán học, thống kê để dự đoán, tối ưu hóa
* DSS hỗ trợ người dùng tương tác với dữ liệu (what-if analysis), mô phỏng, dự báo, và cung cấp báo cáo phân tích sâu cho quản lý

*\*Bảng tổng quan hiệu suất hàng tháng dành cho CEO - EIS (Executive Information System)*=> Giải thích**:**

* CEO cần theo dõi toàn bộ hoạt động toàn hệ thống và ra quyết định chiến lược
* EIS được thiết kế cho lãnh đạo cấp cao: tóm tắt, trực quan, khả năng drill-down tối thiểu, truy cập nhanh.
* EIS tập trung vào chiến lược và xu hướng dài hạn. Giao diện thân thiện, dễ hiểu trong vài giây.

**Bài 2:**

***-*** *A: Phần mềm quản lý điểm cho trường cấp 2, yêu cầu rõ ràng*

=> Mô hình: Waterfall

* Yêu cầu rõ ràng, ít thay đổi trong quá trình phát triển.
* Phần mềm quản lý điểm có quy trình nghiệp vụ quen thuộc, có thể xác định rõ yêu cầu ngay từ đầu, tài liệu đầy đủ
* Waterfall phù hợp khi quy trình có các giai đoạn tuần tự các pha SDLC (yêu cầu – thiết kế - lập trình – kiểm thử - triển khai – bảo trì) và ít rủi ro thay đổi.
* Dễ quản lý tiến độ, đặc biệt với dự án nhỏ hoặc trung bình trong lĩnh vực giáo dục.

***-*** *B: Ứng dụng mobile đặt lịch khám bệnh, yêu cầu linh hoạt*

=> Mô hình: Agile

* Yêu cầu linh hoạt, thay đổi theo phản hồi của người dùng (bệnh nhân, bác sĩ).
* Agile cho phép phát triển theo vòng lặp (iteration/sprint), giúp nhóm nhanh chóng ra phiên bản thử nghiệm, thu thập phản hồi và cải thiện.
* Agile phù hợp để đảm bảo tính linh hoạt và khả năng đáp ứng thay đổi.
* Giúp nhóm phát triển và khách hàng tương tác thường xuyên, điều chỉnh tính năng theo nhu cầu thực tế.

***-*** *C: Hệ thống ngân hàng điện tử có bảo mật và độ phức tạp cao*

=> Mô hình: Spiral (xoắn ốc)

* Hệ thống có độ phức tạp cao, yêu cầu bảo mật nghiêm ngặt, rủi ro lớn nếu sai sót.
* Mô hình Spiral tập trung vào đánh giá và kiểm soát rủi ro ở mỗi vòng phát triển, giúp giảm thiểu lỗi và đảm bảo an toàn.
* Phù hợp cho dự án lớn, quan trọng, cần kiểm thử và đánh giá kỹ ở từng vòng trước khi mở rộng.

**Bài 3:**

* *Con người:*

+ Người sử dụng hệ thống (gồm khách hàng, tài xế, chủ quán và quản trị hệ thống)

+ Mỗi nhóm người dùng có vai trò riêng: khách đặt món, quán nhận đơn và chuẩn bị, tài xế giao hàng, quản trị giám sát và xử lí sự cố

* *Dữ liệu*

+ Là thông tin được thu nhập và xử lí (gồm thông tin người dùng, danh sách món, giá cả, địa chỉ giao hàng, trạng thái đơn, đánh giá – phan hồi, và dữ liệu thanh toán)

+ Hệ thống lưu trữ và cập nhật liên tục

* *Quy trình*

+ Là cách thức hoạt động của hệ thống

+ Chuỗi hoạt động từ khi: khách đặt món -> quán xác nhận -> tài xế nhận đơn -> giao hàng -> khách nhận món và thanh toán

+ Ngoài ra, còn có quy trình xử lí khiếu nại, hoàn tiền, đánh giá…

* *Phần mềm*

+ Là công cụ để con người thao tác (ứng dụng mobile, website…, ứng dụng riêng cho tài xế, quán ăn cùng hệ thống quản lí)

+ Giúp người dùng tương tác theo dõi đơn, thanh toán trực tuyến…

* Phần cứng

+ Hạ tầng vật lí để hệ thống vận hành

+ Bao gồ điện thoại thông minh, máy tính, hệ thống mạng…

**Bài 4: Ứng dụng điểm danh**

*-Planning*

Xác định mục tiêu của ứng dụng, phạm vi dự án, đối tượng sử dụng, dự trù chi phí, thời gian, nhân sự

*-Analysis*

Thu nhập yêu cầu từ người dùng: Cách điểm danh, thông tin cần thiết, quyền truy cập, các báo cáo…, xác định các yêu cầu

*-Design*

Thiết kế giao diện người dùng, cơ sở dữ liệu (sinh viên, lớp học, các lớp học, điểm danh…) và kiến trúc hệ thống (ứng dụng web/mobile, server…)

*-Implementation*

Viết mã nguồn cho ứng dụng, phát triển các chức năng, lưu dữ liệu…

*-Deployment & Maintenance*

+ Triển khai ứng dụng lên máy chủ hoặc kho ứng dụng, hướng dẫn người dùng sử dụng

+ Bảo trì, cập nhập tính năng, xử lí sự cố trong quá trình vận hành

**Bài 5: Hệ thống điểm danh sinh viên bằng QR code**

*\*Planning*

-Mục tiêu:

Xây dựng hệ thống điểm danh bằng mã QR nhằm tự động hóa quá trình điểm danh, giảm gian lận và giúp quản lí dữ liệu hiệu quả hơn

-Công việc:

+ Xác định phạm vi dự án

* Quản lí lớp, các buổi học, danh sách sinh viên
* Tích hợp tính năng tạo mã QR cho mỗi buổi học, cho phép sinh viên điểm danh
* Phòng đào tạo có thể xem thống kê, xuất báo cáo điểm danh

+ Xác định người dùng chính

Giảng viên, sinh viên, phòng đào tạo…

+ Dự kiến công nghệ

Ứng dụng, Backend, hệ thống mã và xác thực

+ Thời gian & nhân sự

Dự kiến số người, trong khoảng bao lâu

*\*Requirement Analysis*

-Mục tiêu:

Thu nhập, phân tích và mô tả rõ các yêu cầu chức năng và phi chức năng của hệ thống

-Yêu cầu chức năng

+ Giảng viên

* Tạo lớp học, các buổi học
* Sinh mã QR chứa thông tin buổi học
* Xem danh sách sinh viên đã điểm danh

+ Sinh viên

* Đăng nhập vào ứng dụng
* Quét mã điểm danh
* Xem lịch sử điểm danh

+ Phòng đào tạo

* Quản lí tài khoản giảng viên và sinh viên
* Theo dõi thống kê tỉ lệ điểm danh từng lớp, từng sinh viên
* Xuất báo cáo theo kì, theo môn

-Yêu cầu phi chức năng

+ Giao diện thân thiện, dễ sử dụng

+ Bảo mật thông tin sinh viên

+ Thời gian phản hồi sau khi quét

* Sơ đồ phù hợp: Use Case Diagram – thể hiện mối quan hệ giữa các tác nhân và các chức năng của hệ thống

*\*System Design*

-Mục tiêu:

Chuyển yêu cầu thành mô hình kĩ thuật và kiến trúc hệ thống cụ thể

-Công việc

+ Thiết kể kiến trúc tổng thể (client, server, db…)

+ Thiết kế cơ sở dữ liệu

+ Thiết kế giao diện

* Sơ đồ phù hợp

-Class Diagram: mô tả cấu trúc các lớp trong hệ thống và mối quan hệ giữa chúng

-Sequence Diagram: mô tả trình tự các thao tác khi sinh viên quét mã điểm danh

**Bài 6**

*\* A – Use Case Diagram*

=>Sơ đồ mô tả người dùng và chức năng họ có thể thực hiện trên hệ thống -> Phù hợp để thể hiện chức năng của ứng dụng học tiếng Anh (như đăng nhập, làm bài, xem kết quả…)

*\* B – Class Diagram*

=> Sơ đồ mô tả các lớp dữ liệu, phương thức và mối quan hệ giữa các lớp -> Phù hợp khi mô tả cấu trúc đối tượng như người dùng, khóa học, bài học

*\* C – Activity Diagram*

=> Sơ đồ biểu diễn luồng công việc hoặc quy trình -> Phù hợp để mô tả tiến trình của học viên

*\* D – Deployment Diagram*

=> Sơ đồ mô tả cấu trúc vật lý của hệ thống - bao gồm các node (máy chủ, thiết bị) và thành phần phần mềm triển khai trên đó.

*\* E – Sequence Diagram*

=> Sơ đồ mô tả thứ tự các thông điệp, hành động được trao đổi giữa các đối tượng theo thời gian -> Phù hợp hợp khi mô tả tương tác nộp bài giữa học viên và hệ thống.

**Bài 7**

*\*Planning*

-Mục tiêu: Xác định phạm vi, mục đích, thời gian và nguồn lực dự án.

-Việc cần làm:

+ Thu thập thông tin về nhu cầu từ trung tâm y tế.

+ Xác định đối tượng sử dụng: người dân, nhân viên y tế, quản trị hệ thống.

+ Lập kế hoạch tổng quan về tiến độ, chi phí, nhân sự, rủi ro.

+ Đặt mục tiêu: Hệ thống cho phép người dân đăng ký tiêm online, trung tâm xác nhận, theo dõi lịch sử tiêm.

\*Requirement Analysis

Việc cần làm:

- Phỏng vấn nhân viên y tế và người dân để xác định nhu cầu.

- Phân loại yêu cầu:

+ Yêu cầu chức năng: đăng ký tiêm, xác nhận lịch, theo dõi người tiêm/chưa tiêm.

+ Yêu cầu phi chức năng: bảo mật thông tin cá nhân, giao diện dễ sử dụng, tốc độ phản hồi nhanh.

- Viết tài liệu SRS (Software Requirements Specification) để thống nhất với các bên liên quan.

\*System Design

-Việc cần làm:

+ Thiết kế kiến trúc hệ thống (web/mobile app kết nối cơ sở dữ liệu trung tâm).

+ Thiết kế CSDL (bảng NgườiDân, LịchTiêm, TrạngTháiTiêm, NhânViênYTe...).

+ Thiết kế giao diện người dùng: trang đăng ký, xác nhận, thống kê.

+ Sử dụng UML diagrams như Use Case, Class, Sequence để mô tả hoạt động.

\*Implementation

-Mục tiêu: Xây dựng hệ thống dựa trên bản thiết kế.

-Việc cần làm:

+ Phân chia nhiệm vụ cho các lập trình viên (frontend, backend, database).

+ Viết mã và kết nối hệ thống

+ Kiểm tra tính năng trong nhóm phát triển.

+ Đảm bảo tuân thủ tiêu chuẩn bảo mật dữ liệu y tế.

\*Testing

Việc cần làm:

+ Thực hiện các loại kiểm thử:

+ Sửa lỗi, cải thiện trải nghiệm.

\*Deployment & Maintenance

-Mục tiêu: Đưa hệ thống vào hoạt động và duy trì ổn định.

-Việc cần làm:

+ Triển khai hệ thống lên máy chủ thật.

+ Đào tạo nhân viên y tế cách sử dụng hệ thống.

+ Theo dõi hiệu suất và phản hồi người dùng.

+ Thường xuyên cập nhật, vá lỗi, thêm tính năng mới (ví dụ: nhắc lịch tiêm qua SMS, thống kê tỷ lệ tiêm).

**Bài 8: Nền tảng học trực tuyến dành cho trung tâm ngoại ngữ**

1.Tác nhân chính

|  |  |
| --- | --- |
|  | **Chức năng chính** |
| **Học viên** | - Đăng ký tài khoản và khóa học - Truy cập và học bài trực tuyến - Làm bài kiểm tra, xem điểm số - Theo dõi tiến độ học tập |
| **Giảng viên** | |  | | --- | |  |  |  | | --- | | - Tạo và quản lý khóa học, bài giảng - Ra bài tập, chấm điểm học viên - Quản lý danh sách học viên của khóa học | |
| **Quản trị viên** | - Quản lý người dùng (học viên, giảng viên) - Quản lý khóa học, phân quyền truy cập - Theo dõi báo cáo và thống kê hoạt động học tập |

2.Phân loại hệ thống

|  |  |  |
| --- | --- | --- |
| **Chức năng** | **Loại hệ thống** | **Ghi chú** |
| Học viên đăng kí, học và làm bài | TPS | Vì hệ thống xử lý các giao dịch thường xuyên như đăng ký khóa học, nộp bài, ghi nhận điểm. |
| Giảng viên và admin xem báo cáo, thống kê | MIS | Tổng hợp dữ liệu từ TPS để tạo báo cáo về tiến độ học, tỷ lệ hoàn thành khóa, số lượng học viên… |
| Admin phân quyền, đưa ra quyết định quản trị | DSS | Hỗ trợ nhà quản lý phân tích dữ liệu học viên – giảng viên để ra các quyết định liên quan |

3.Đề xuất mô hình phát triển: **Mô hình Agile**

* Hệ thống học trực tuyến có nhiều chức năng phức tạp và thay đổi linh hoạt
* Agile cho phép phát triển từng phần nhỏ, nhận phản hồi sớm từ giảng viên/học viên để điều chỉnh kịp thời.
* Phù hợp với môi trường trung tâm ngoại ngữ, nơi yêu cầu trải nghiệm người dùng và cải tiến tính năng liên tục.
* Dễ mở rộng (thêm tính năng thống kê, gamification, AI đánh giá học viên,...).

4.Sơ đồ UML sử dụng

*-Use Case Diagram*

Mô tả các tác nhân (học viên, giảng viên, admin) và chức năng mà họ có thể thực hiện trong hệ thống.

*-Class Diagram*

Mô tả các lớp dữ liệu (HọcVien, GiangVien, KhoaHoc, BaiHoc, DiemSo, BaoCao…) và mối quan hệ giữa chúng.

*-Sequence Diagram*

Minh họa luồng tương tác giữa học viên, giảng viên và hệ thống trong các quy trình như “Đăng ký khóa học” hoặc “Chấm điểm bài tập”.

**Bài 9**

**1.Các tác nhân**

*- Khách hàng (Customer)*

+ Tạo đơn (địa chỉ, trọng lượng, loại hàng)

+ Thanh toán / chọn phương thức thanh toán

+ Theo dõi trạng thái đơn theo thời gian thực

+ Hủy/đổi đơn, gửi khiếu nại

*- Nhân viên vận chuyển / Shipper (Driver)*

+ Nhận/nhận lệnh giao (accept assignment)

+ Cập nhật trạng thái: picked up, in transit, delivered, failed delivery

*- Quản lý vận hành*

+ Giám sát hiệu suất khu vực

+ Quản lý user, cấu hình phí vận chuyển, vùng vận hành

*- CEO / Ban lãnh đạo*

Xem dashboard tổng quan: giao dịch, vùng hoạt động, on-time delivery rate, revenu

**2.Phân loại hệ thống**

*TPS = xử lý giao dịch hàng ngày*

*MIS = báo cáo quản lý*

*DSS = hỗ trợ ra quyết định (phân tích, mô phỏng)*

*EIS = dashboard điều hành cho lãnh đạo.*

|  |  |  |
| --- | --- | --- |
|  | **Loại hệ thống** | **Ghi chú** |
| Khách hàng tạo đơn, thanh toán, cập nhật địa chỉ | TPS | Giao dịch thường xuyên, yêu cầu tính chính xác và ghi nhận ngay |
| shipper cập nhật trạng thái | TPS | Cập nhật trạng thái theo thời gian thực, là nghiệp vụ giao dịch. |
| Theo dõi đơn theo thời gian thực | TPS + MIS | Tracking là giao dịch; dữ liệu tracking được tổng hợp thành báo cáo. |
| Báo cáo hằng ngày / tuần về số đơn, tỉ lệ giao thành công | MIS | MIS tổng hợp dữ liệu TPS thành báo cáo vận hành tiêu chuẩn |
| Phân tích nguyên nhân tắc nghẽn, tối ưu phân bổ shipper, what-if (thêm tuyến, thay đổi nhân lực) | DSS | Cần mô phỏng, phân tích, tối ưu — hỗ trợ ra quyết định. |
| Dự báo nhu cầu theo vùng/giờ để phân bổ nguồn lực | DSS | Mô hình dự báo, phân bổ tài nguyên. |
| Dashboard CEO: giao dịch tổng, vùng hoạt động, on-time rate, revenue trend | EIS | Tóm tắt KPI chiến lược, view cấp cao, drill-up ít, trực quan nhanh |

3.Mô hình phát triển: **Agile**

- Phức tạp và nhiều thành phần chức năng (mobile app khách hàng, driver app, backend services, dashboard…)

- Yêu cầu thay đổi thường xuyên (chính sách phí, cấu hình vùng,…)

- Yêu cầu khả năng mở rộng cao

4.Sơ đồ UML

**- Use Case Diagram**

+ Mục đích: Xác định actors và các chức năng hệ thống ở mức yêu cầu

+ Khi dùng: Giai đoạn phân tích yêu cầu để thương thảo với stakeholder.

**- Class Diagram**

+ Mục đích: Mô tả mô hình dữ liệu chính: Order, Customer, Shipment, Driver,.. quan hệ giữa chúng.  
+ Khi dùng: Thiết kế database và interface giữa services.

**- Sequence Diagram**

+ Mục đích: Minh hoạ trình tự tương tác khi khách tạo đơn → hệ thống xác nhận → dispatcher phân công → driver nhận → cập nhật trạng thái → giao hàng thành công

+ Khi dùng: Mô tả luồng thông điệp giữa client, API gateway, order service, notification

**- Deployment Diagram**

+ Mục đích: Mô tả kiến trúc triển khai

+ Khi dùng: Giai đoạn thiết kế hạ tầng & triển khai