**MÔN KIẾN TRÚC & THIẾT KẾ PHẦN MỀM**

**TÀI LIỆU 35 CÂU HỎI LÝ THUYẾT**

**2.2 Phần lý thuyết: chọn 1 trong số các câu hỏi về các nguyên lý kỹ thuật & công nghệ trong thiết kế phần mềm.**

# Câu 1:

a. Khi xây dựng kiến trúc & thiết kế phần mềm, phải đảm bảo những yều cầu gì? vì sao? minh họa thực tế.

b. Giả sử khi thiết kế kiến trúc phần mềm uis, cần quan tâm những công nghệ hiện tại nào? sử dụng vào chức năng gì ?

**1a:** Thiết kế kiến trúc phần mềm phải phù hợp với thực trạng phát triển của kĩ thuật và công nghệ, nghĩa là không xây dựng thiết kế một hệ thống phần mềm quá lạc hậu, không còn phù hợp nữa, hoặc sử dụng công nghệ, yêu cầu kĩ thuật quá mới, trình độ kĩ thuật công nghệ mặt bằng chung hiện tại chưa đáp ứng kịp. Dù là quá lạc hậu hay quá mới thì hậu quả vẫn là phần mềm làm ra không có tính ứng dụng, lãng phí thời gian và chi phí phát triển.

**VD:** Xu hướng phát triển phần mềm hiện tại là theo quy mô toàn cầu, đồng nghĩa khả năng tiếp cận ứng dụng phải cao, do đó trừ các phần mềm đặc thù, cần nhiều tài nguyên xử lý thì các công việc quản lý, văn phòng, mua bán hàng cơ bản đều phát triển dựa trên website và mobile. Nếu phát triển phần mềm cho các công việc đó trên desktop thì sẽ rất kén người dùng, được coi như là lạc hậu, ngược lại, các công việc đặc thù như chỉnh sửa ảnh chuyên nghiệp, lập trình,… thì công nghệ, kĩ thuật, đặc thù công việc hiện tại thể đáp ứng việc đưa chúng lên website hay mobile.

**1b:** Khi thiết kế phần mềm uis, cần phải quan tâm đến 3 vấn đề chính là khả năng chịu tải tốt, tính bảo mật cao, thanh toán online. Khả năng chịu tải tốt là để chịu được lượng truy cập lớn, cùng lúc vào các thời điểm đăng ký môn họ. Tính bảo mật cao là do đây là các dữ liệu quan trọng, nếu bị rò rỉ hoặc chỉnh sửa trái phép thì sẽ dẫn đến hậu quả rất lớn. Thanh toán online dùng vào việc thanh toán học phí sau khi đăng kí môn học trở nên tiện lợi dễ dàng và nhanh chóng hơn.

# Câu 2:

Khi phân tích và thiết kế kiến trúc hệ thống phần mềm sử dụng trên qui mô toàn cầu phát sinh những vấn đề gì cần quan tâm? Minh họa thực tế

Khi phát triển phần mềm trên quy mô toàn cầu sẽ phát minh những vấn đề sau đây:

* Yêu cầu tính bảo mật cao, vì tất cả mọi người ai cũng có thể truy cập được nên khả năng bị hack, mất an toàn là rất cao.

**VD:** Sử dụng phương thức https thay cho http để mã hóa dữ liệu gửi và nhận.

* Cần có dịch vụ thanh toán điện tử nếu phần mềm có chức năng thanh toán để tăng tính tiện dụng, vì người sản xuất và người dùng phần mềm không biết nhau, và có thể cách nhau một khoảng cách rất xa.

**VD:** Triển khai các hình thức thanh toán toàn cầu bằng thẻ như VISA, Mastercard, JCB,…

* Phần mềm phải có khả năng chịu tải lớn, cùng với đó là các giải pháp giúp tối ưu hóa truy cập.

**VD:** Sử dụng mã CAPTCHA để loại bỏ các yêu cầu spam. Sử dụng MongoDB thay cho các CSDL quan hệ để tăng tốc độ truy vấn cho các dữ liệu lớn nhưng ít chỉnh sửa như dữ liệu thống kê.

* Cần phải có hệ thống đa ngôn ngữ cho phần mềm, cũng như quan tâm đến các vấn đề về văn hóa, tôn giáo liên quan đến pháp luật nhiều quốc gia.

**VD:** Tiếng Anh là ngôn ngữ tất yếu, phải có trong hệ thống đa ngôn ngữ khi triển khai phần mềm mang tính toàn cầu, vì đó được coi là ngôn ngữ toàn cầu.

* Hệ thống máy tính mở, không giới hạn ở một vài máy tính, nghĩa là phần mềm phải có khả năng mở rộng cao, tham gia phát triển bởi nhiều cá nhân, tổ chức, sử dụng open source để tiết kiệm chi phí và thời gian phát triển. Tuy vậy, cũng cần quan tâm đến các vấn đề pháp lý, sở hữu trí tuệ đi kèm khi sử dụng các giải pháp mở rộng nêu trên.

# Câu 3:

Các giai đoạn cơ bản trong tiến trình phát triển hệ thống phần mềm? Các bước cụ

thể trong các giai đoạn đó? Minh họa đơn giản cho mỗi bước?

Có 2 giai đoạn cơ bản trong quá trình phát triển hệ thống phần mềm:

1. Xây dựng kiến trúc hệ thống phần mềm

* Phân tích kiến trúc phần mềm
* Thiết kế kiến trúc phần mềm

1. Thiết kế hệ thống phần mềm

Bao gồm các nội dung thiết kế sau:

* Giao diện
* Chức năng xử lý
* Cơ sở dữ liệu
* Giao tiếp ngoài (QR Code, Bar code, thanh toán điện tử,…)
* Hệ thống bảo mật (Chữ ký số, mã hóa dữ liệu, mật khẩu,…)
* Hệ thống trợ giúp (Tooltip, Báo lỗi, dialog thông báo,…)
* Cài đặt và chuyển giao (Cài đặt ở đâu, client hay server. Triển khai lên nền tảng nào, như thế nào. VD: mobile app, lên app store, CH play. Web app, thuê domain,…)

Trong giai đoạn 2 không bắt buộc phải thiết kế đầy đủ cả 8 nội dung và thứ tự nêu trên, mà phải xem xét vào từng hoàn cảnh, trường hợp cụ thể, cân nhắc chi phí, thời gian,…

# Câu 4:

Phân tích kiến trúc hệ thống phần mềm bao gồm những công việc cơ bản nào? Minh họa thực tế?

Để đảm bảo tính chất bảo mật (security) & an toàn (safety) của Hệ thống phần mềm, thường phải có Hệ thống kiến trúc phần mềm như thế nào? Và với Hệ thống kến trúc phần mềm đó, cần phải thiết kế nhưng chức năng gì để thống nhất thông tin và an toàn Hệ thống phần mềm? Minh họa thực tế với uis.ptithcm.edu.vn

Phân tích kiến trúc hệ thống phần mềm bao gồm những công việc:

1. Về qui mô và giới hạn phạm vi của hệ thống phần mềm
2. Về hình thức sử dụng phần mềm
3. Về cơ sở nền tảng công nghệ
4. Các dạng mô hình kiến trúc phần mềm và yêu cầu của từng mô hình
5. Xác định đối tượng sử dụng & các quyền bảo mật tương ứng

Để đảm bảo tính an toàn và bí mật, khi thiết kế kiến trúc và các hệ thống phần mềm, có thể

-Xác định các dạng app có thể có của hệ thống (VD Dạng web app và mobile app)

-Dữ liệu được lưu song song, vừa ở trên điện toán đám mây, vừa ở máy tính cục bộ

Khi đó, phải thiết kế các chức năng

* Phải có các phương thức đồng bộ dữ liệu giữa cục bộ và đám mây thích hợp (Đồng bộ định kì, đồng bộ ngoài giờ cao điểm, đồng bộ theo yêu cầu)
* Đảm bảo phân biệt người dùng trên điện toán đám mây và người dùng trên hệ thống máy tính cục bộ để giải quyết vấn đề bảo mật. Đối với người dùng trên đám mây thì buộc phải đảm bảo tính chất bảo mật cao nhất, để tránh trường hợp bị đánh cắp thông tin.

# Câu 5 (Hơi giống câu 3, nhưng thêm ý về quan hệ)

a. Mối quan hệ giữa Kiến trúc phần mềm và Thiết kế phần mềm? Minh họa thực tế?

b. Có cần phải Thiết kế kiến trúc phần mềm cho tất cả các trường hợp phần mềm hay không, (Chỉ Thiết kế Kiến trúc phần mềm đối với những trường hợp phần mềm như thế nào?) Vì sao?

*Sau khi hoàn thành việc Xây dựng Kiến trúc phần mềm, ta sẽ hình thành được rất nhiều các component của phần mềm, với mỗi component đó ta sẽ tiến hành Thiết kế hệ thống phần mềm cho từng component đó theo 8 nội dung thiết kế sau:*

* *Giao diện*
* *Chức năng xử lý*
* *Cơ sở dữ liệu*
* *Giao tiếp ngoài (QR Code, Bar code, thanh toán điện tử,…)*
* *Hệ thống bảo mật (Chữ ký số, mã hóa dữ liệu, mật khẩu,…)*
* *Hệ thống trợ giúp (Tooltip, Báo lỗi, dialog thông báo,…)*
* *Cài đặt và chuyển giao (Cài đặt ở đâu, client hay server. Triển khai lên nền tảng nào, như thế nào. VD: mobile app, lên app store, CH play. Web app, thuê domain,…)*

*Trong giai đoạn thiết kế hệ thống phần mềm, không bắt buộc phải thiết kế đầy đủ cả 8 nội dung và thứ tự nêu trên, mà phải xem xét vào từng hoàn cảnh, trường hợp cụ thể, cân nhắc chi phí, thời gian.*

**5a:** Mối quan hệ: Với 1 hệ thống phần mềm ta chỉ xây dựng kiến trúc 1 lần (thực hiện 1 lần), sau khi xây dựng xong sẽ hình thành nhiều component, với mỗi component đó ta sẽ tiến hành 8 nội dung thiết kế nêu trên (thực hiện nhiều lần).

**5b:** Phân tích khi quy mô của hệ thống phần mềm tương, của công ty đối lớn thì mới cần xây dựng kiến trúc.

# Câu 6:

Các yêu cầu và nguyên tắc cơ bản về thiết kế phần mềm? Minh họa?

***Nguyên tắc***

1. Lấy người dùng làm trung tâm, theo nguyên tắc lấy ý kiến, thông qua tham vấn kĩ thuật để thống nhất yêu cầu của phần mềm đó.

**VD:** Người dùng yêu cầu thực hiện chức năng thoát chương trình chạy trên website như trên win app, về mặt kĩ thuật thì không thể thực hiện chức năng này giống trên win app, vì vậy cần phải tham vấn kĩ thuật, chỉnh sửa lại, sử dụng chức năng đăng nhập, đăng xuất.

1. Bảo đảm người dùng sử dụng triệt để các chức năng của phần mềm nhằm tránh tình trạng người dùng không biết, không nhớ các tính năng phần mềm đã có. Để làm được việc này ta cần có một hệ thống trợ giúp hiệu quả

**VD:**

Hệ thống trợ giúp tổng quan, giới thiệu các tính năng của chương trình: HelpProvider

Hỗ trợ, gợi ý tại mỗi bước nhập liệu để hạn chế sai: Tooltip, hint

Vị trí, tiến độ trong nghiệp vụ đang thao tác: Sử dụng giao diện, đồ họa.

1. Hỗ trợ người dùng có bản quyền

**VD của Tú:** Đối với người dùng đã mua bản quyền thì sẽ có các hỗ trợ, giải đáp thắc mắc về phần mềm nhanh chóng, trực tiếp với nhân viên kĩ thuật. Mở khóa các chức năng độc quyền, nâng cao.

**VD của Phong:** Đối với người dùng miễn phí sẽ hạn chế nhiều tính năng quan trọng hoặc cho họ dùng thử trong 1 thời gian giới hạn để người dùng đưa ra quyết định

1. Hiệu quả kinh tế và ứng dụng, bao gồm công sức và chi phí để làm ra phần mềm phải nhỏ hơn kết quả đạt được.

***Yêu cầu:***

* Cho phép người dùng chọn 1 trong nhiều phương án để thực hiện 1 công việc nào đó

**VD của Tú:**

Với công việc tìm kiếm 1 đoạn văn bản

Đối với người dùng thành thạo: Sử dụng hotkey [Ctrl + F]

Đối với người mới sử dụng: Menu ⇒ Home ⇒ Find

**Ví dụ của Phong:**

Với việc gửi một hình ảnh

Đối với người dùng thành thạo: kéo và thả ảnh muốn gửi vào

Đối với người dùng mới: chọn nút “gửi hình ảnh”

* Thiết kế phần mềm phải đảm bảo tập trung, ưu tiên cho người sử dụng thành thạo.
* Hướng đơn giản cho người dùng, tránh gây rối, không nên sử dụng quá nhiều màu sắc và các màu sắc gây khó chịu, áp lực cho người sử dụng

**VD của Tú:** Màu gây khó chịu, áp lực: màu đỏ, chỉ thường được dùng trong cảnh báo

**Ví dụ của Phong:** Dùng bộ icon tối giản thay vì các icon sặc sỡ, nhiều chi tiết

* Có sự thay đổi độ phá, tránh gây nhàm chán cho người dùng lâu dài
* Đảm bảo tâm lý tốt cho người dùng

# Câu 7:

a) Thiết kế các hình thức thực hiện một thao tác nghiệp vụ trên phần mềm, ưu tiên cho những người dùng nào? Vì sao? Minh họa thực tế?

b) Xét thao tác in văn bản trong phần mềm Word, theo anh chị, nên thiết kế những phương pháp nào để thực hiện thao tác trên? Phương pháp nào đc ưu tiên hơn? Vì sao ?

c) Xét phần mềm uis.ptithcm.edu.vn với thao tác đăng nhập, theo anh chị nên thiết kế những phương pháp nào để người dùng lựa chọn và thực hiện? Phương pháp nào ưu tiên hơn? Vì sao?

**7a:**

Phải ưu tiên người dùng thành thạo. Bởi vì đối người dùng mới, ta sẽ thiết kế các hệ thống trợ giúp riêng giúp họ làm quen với phần mềm. Thêm vào đó, người mới sử dụng sau khi trải qua một thời gian sử dụng thì đều sẽ trở thành người dùng thành thạo

⇒ Nếu ta thiết kế thao tác mà ưu tiên người mới bắt đầu sẽ dẫn đến sau này người dùng cảm thấy khó chịu, vướng víu với các hướng dẫn dày đặc, và thao tác làm việc quá chậm khi phải chọn lần lượt từ menu.

**VD của Tú:**

Với công việc tìm kiếm 1 đoạn văn bản

Đối với người dùng thành thạo: Sử dụng hotkey [Ctrl + F]

Đối với người mới sử dụng: Menu ⇒ Home ⇒ Find

**Ví dụ của Phong:**

Với việc gửi một hình ảnh

Đối với người dùng thành thạo: kéo và thả ảnh muốn gửi vào

Đối với người dùng mới: chọn nút “gửi hình ảnh”

**7b:**

Đối với người dùng mới bắt đầu Ribbon Menu ⇒ File ⇒ Print

Đối với người dùng thành thạo: Sử dụng hotkey [Ctrl + P]

Phương pháp sử dụng hotkey sẽ được ưu tiên hơn, vì nó nhanh hơn đáng kể so với phương pháp của người mới bắt đầu, tuy vậy ta vẫn thiết kế 2 cách để họ làm quen với phần mềm, hoặc người dùng thành thạo quên hotkey, tránh việc chương trình quá tối nghĩa, khó hiểu.

**7b) Tiếp**

**Ví dụ của Phong:**

Cho phép người dùng có thể gán thêm nút để người dùng tự quyết định tổ hợp phím bấm mong muốn

Tạo một chức năng học thói quen của người dùng. Chương trình sẽ hiện chức năng Print nên đầu mục lựa chọn nếu thấy người dùng hay sử dụng chức năng này

**7c:**

Đối với người mới bắt đầu: Nhấp chọn ô “Tên đăng nhập” ⇒ Điền thông tin ⇒Nhấp chọn ô “Mật khẩu” ⇒ Điền thông tin ⇒Nhấp chọn “Đăng nhập”

Đối với người dùng thành thạo: Nhấp chọn ô “Tên đăng nhập ⇒ Điền thông tin ⇒ Tab ⇒ Điền thông tin ⇒ Enter

Phương pháp của người dùng thành thạo sẽ được ưu tiên hơn, vì gần như tay của người dùng không phải chuyển qua lại giữa chuột và bàn phím, giúp việc nhập thông tin nhanh hơn đáng kể, tuy nhiên ta vẫn phải có phương pháp truyền thống đầu để tránh chương trình trở nên quá tối nghĩa, khó sử dụng với người mới.

# Câu 8:

Các bước tương tác cơ bản của người sử dụng trên hệ thống phần mềm? Minh họa thực tế? Có bắt buộc mọi thao tác có đầy đủ 4 bước (VD)? Bước nào quan trọng nhất? Vì sao? Minh họa?

Các bước thao tác của người dùng: 4 bước

**B1:** chọn lệnh / thao tác cần thực hiện

**VD:** Đăng ký tài khoản

**Ví dụ của Phong:** thay đổi thông tin tài khoản

**B2:** cung cấp các thông tin cần thiết để phần mềm thực hiện lệnh / thao tác đó

**VD:** Điền các thông tin cần thiết của tài khoản (email, mật khẩu, họ tên,…)

**Ví dụ của Phong:** chọn hình ảnh để làm ảnh đại diện

**B3:** Ra lệnh cho phần mềm thực hiện lệnh / thao tác cần thiết sau khi cung cấp đầy đủ thông tin

**B4:** Người dùng tiếp nhận kết quả trả về từ lệnh / thao tác nêu trên, có 2 khả năng

* Đạt yêu cầu
* Không đạt yêu cầu, lỗi, phần mềm hỗ trợ người dùng khắc phục lỗi

Chú ý:

1. Các bước nêu trên là 1 trường hợp kịch bản khai thác sử dụng phần mềm của người dùng trong trường hợp đúng nhất tuy nhiên trên thực tế người dùng có thể thực hiện theo kịch bản khác.

**VD:**

* Đang điền thông tin đăng ký (B2) thì chọn lệnh yêu cầu đăng nhập (B1).
* Mất kết nối Internet khi gửi thông tin đăng ký (B3)

**Ví dụ của Phong:**

* Đang chọn tải ảnh đại diện lên thì mất internet
* Đang lưu thông tin tài khoản thì ứng dụng vị đứng, văng ra ngoài

1. Không nhất thiết tất cả các thao tác đều có đầy đủ 4 bước nêu trên và không nhất thiết theo trình tự:

**VD**:

* Xóa thông tin vật tư: Chọn vật tư muốn xóa (B2) ⇒ Ra lệnh xóa (B3) ⇒ Kết quả (B4)
* Sửa thông tin vật tư: Chọn vật tư muốn sửa (B2) ⇒ Yêu cầu sửa (B1) ⇒ Điền thông tin sửa (B3) Kết quả ⇒ (B4)

**Ví dụ của Phong:**

* Sửa tên hiển thị của tài khoản: sau khi nhập thông tin => thoát => hệ thống tự động lưu thay đổi
* Word tự động lưu lại đoạn văn bản người dùng đang làm việc mà không cần nhấn nút “Save”

1. B3 là bước quan trọng nhất, quyết định việc chương trình đó có chạy đúng hay không, khi lập trình phải suy xét các kịch bản lỗi có thể xảy ra.

# Câu 9:

Khi thiết kế hệ thống phần mềm WebApp cần quan tâm vấn đề gì? Minh họa thực tế? Khi xác định múi giờ thao tác của người dùng ta chọn múi giờ phía Client hay Server? Vì sao?

1. Cường độ truy xuất phần mềm là giống nhau vào tất cả thời điểm trong ngày, do phạm vi toàn cầu, khác với WinApp.
2. Xác định múi giờ cập nhật / xử lý trên hệ thống phần mềm sao cho phù hợp trong các giao dịch của người dùng, tùy phần mềm WebApp.
3. Thông thường, ngày giờ lưu trong database nên lưu ở UTC + 0 (Ngày giờ chuẩn), để loại bỏ hoàn toàn sự phụ thuộc vào múi giờ. Từ đó, tùy thuộc vào múi giờ của người dùng mà chương trình tự động đổi về tương ứng

**VD:** Thông tin thời điểm đặt hàng trên WebApp bán hàng trực tuyến, của một món hàng được mua vào 13/09/2022 8:00 A.M (UTC + 0)

* Phần quản lý **tất cả** đơn hàng (Nhân viên bán hàng sử dụng) sẽ lưu ngày giờ **bán** theo múi giờ địa điểm của nhân viên bán hàng là UTC – 3 sẽ là 13/09/2022 5:00 A.M (UTC -3)
* Phần quản lý đơn hàng **cá nhân** của tài khoản (Khách mua hàng sử dụng) sẽ lưu ngày giờ mua theo múi giờ địa điểm của khách mua hàng là UTC + 5 sẽ là 13/09/2022 1:00 P.M (UTC + 5)

**Ví dụ của Phong:**

* Hệ thống chạy ở quốc gia nào thì sẽ lấy thời gian ở quốc gia đó để lưu lại đối với nhân viên
* Với máy chủ khu vực thì sẽ lấy theo chuẩn giờ UTC 0 tại mọi vị trí trên thế giới

1. Phần mềm (WebApp) cũng nên cho chức năng tự động chỉnh múi giờ thủ công theo ý của người dùng
2. Thiết kế giao diện phần mềm đa dạng đa ngôn ngữ ưu tiên ngôn ngữ tại những nơi người dùng nhiều nhất
3. Khi thiết kế các chức năng giao diện, dữ liệu, xử lý nên chú ý phong tục, tập quán, tôn giáo, tín ngưỡng, pháp luật của các cộng đồng người dùng, quốc gia liên quan

**VD của Tú:** Nếu WebApp bán hàng trực tuyến được làm ra chủ yếu được người Việt sử dụng thì nên làm tối thiểu 2 ngôn ngữ là tiếng Việt (vì tệp khách mua hàng đa số dùng tiếng Việt) và tiếng Anh (vì đó là ngôn ngữ chung)

**Ví dụ của Phong:** Nếu ứng dụng chạy ở quốc gia nào thì hỗ trợ tối đa cho ngôn ngữ quốc gia đó. Tuy nhiên, luôn có tùy chọn hỗ trợ tiếng Anh cho khách nước ngoài.

1. Sử dụng kỹ thuật đám mây (cloud computing) và đồng bộ thông tin, dữ liệu.

# Câu 10:

Các nguyên tắc và yêu cầu cơ bản trong thiết kế giao diện phần mềm? Minh họa thực tế?

***Nguyên tắc***

Ngoài các nguyên tắc chung khi thiết kế phần mềm, ta còn có phải chú ý các nguyên tắc sau khi thiết kế giao diện:

1. Lấy người dùng làm trung tâm, mọi yếu tố trện giao diện đều phải xem xét dưới gốc độ của người dùng.
2. Xác định đối tượng người sử dụng giao diện đó, do mỗi người dùng có quyền làm việc riêng nên cần xác định người dùng là ai trước khi thiết kế form. Thêm vào đó, với mỗi form giao diện thì có thể có nhiều người dùng, nhiều vai trò liên quan.

**VD:** Điều chỉnh thông tin sản phẩm trên website: chỉ quản lý cửa hàng mới có quyền, khách mua hàng không được phép truy xuất ⇒ Cần phải có các phương thức bảo mật để ngăn chặn.

**Ví dụ của Phong:** Khi một tính năng mới được thêm vào thì cần phân rõ quyền hạn của từng chức vụ. Ai có thể truy cập và sử dụng chức vụ gì

1. Thiết kế giao diện phải nằm trong tổng thể thiết kế khác như bảo mật, trợ giúp, giao tiếp, cài đặt, kiểm thử,… Vậy nên, chỉ thiết kế giao diện khi kiến trúc phần mềm đã có.

**VD của Tú:** Form điều chỉnh thông tin sản phẩm như ví dụ nêu trên liên quan đến việc thiết kế bảo mật, cần xác định quyền quản lý (có quyền điều chỉnh) và khách hàng (không được phép truy xuất) trước khi tiến hành thiết kế.

**Ví dụ của Phong:** form điều chỉnh thông tin cá nhân nên nằm ở một nửa màn hình và nửa còn lại có thể cho phép người dùng xem trước thay đổi nếu ấn “Lưu”

1. Trước khi thiết kế giao diện cần xác định loại form của giao diện, vì mỗi loại form giao diện có những đặc trưng riêng, ảnh hưởng đến thiết kế. Mỗi chức năng phần mềm có nhiều loại form, vai trò tương đương hoặc không tương đương.

**VD của Tú:**

Mua hàng online trên thương mại điện tử: Web, mobile ⇒ tương đương

Quản lý cửa hàng điều chỉnh thông tin sản phẩm: Windows đảm nhiệm chính, web là phụ

⇒ Không tương đương, do các vấn đề về bảo mật và nền tảng mà Windows sẽ là nơi có nhiều chức năng, tùy chọn hơn, là thông tin chính thức, chốt số liệu.

**Ví dụ của Phong:**

Ứng dụng mua bán thì nên hỗ trợ cả thiết bị di động lẫn máy tính.

Các tính năng dành cho khác hàng trên cả 2 thiết bị nên như nhau

Các tính năng của người bán hàng thì nên hỗ trợ nhiều hơn ở phía máy tính

1. Thiết kế các thành phần của form giao diện cần đóng vai trò người sử dụng, chỉ biết nghiệp vụ, không biết chuyên môn công nghệ.

**VD:** Xét form giao diện quản lý kho hàng, button sau là không hợp lý

Thêm kho – Xóa kho – Sửa kho: Do kĩ thuật cơ sở dữ liệu

⇒ Người dùng không hiểu đúng ý, hiểu xóa kho là dỡ bỏ kho đó một cách vật lý, do không chuyển cơ sở dữ liệu.

1. Ưu tiên người dùng thành thạo

**VD:** không để quá nhiều các hướng dẫn rườm rà, rắc rối chiếm diện tích, khiến người dùng bị khó chịu, do đã hiểu rõ nghiệp vụ.

1. Cần phân tích nghiệp vụ trên giao diện trước khi thiết kế giao diện

⇒ Phải xác định các yêu cầu cụ thể của tất cả hệ thống giao diện và từng giao diện để hỗ trợ việc kiểm thử sau này dễ dàng hơn.

1. Tương đối hóa các giao diện, không nên quá cầu toàn các xử lý trên giao diện.

**VD:** Không báo những lỗi nhỏ nhặt, không cần thiết dẫn đến có quá nhiều thao tác bị cấm, quá nhiều thông báo lỗi dẫn đến khó chịu cho người dùng

1. Cho phép người dùng có nhiều phương án thay đổi giao diện

**VD:**

Option chọn giữa các tùy chọn mà phần mềm đã thiết kế sẵn.

Cusomize: Tự thiết kế, phần mềm không có sẵn

***Yêu cầu***

Yêu cầu của giao diện phần mềm thay đổi tùy thuộc vào loại giao diện, đối tượng người dùng, loại nghiệp vụ, ngoài ra các yêu cầu này còn thay đổi theo thời gian để phù hợp với thời đại. Tuy nhiên có 1 số yêu cầu chuing mà hầu hết các giao diện đều phải đáp ứng.

1. Dễ dùng, không cần học nhiều
2. Gắn liền với quy trình nghiệp vụ, form và report phải phù hợp với biểu mẫu, đảm bảo tính logic, có thể tham vấn người dùng.
3. Đơn giản, màu sắc không gây khó chịu, giao diện thiết kế không tạo cảm giác mất dữ liệu

**VD:**

Tránh các màu nóng như màu đỏ, cam, vàng.

Tránh việc sửa mã làm dòng dữ liệu đó đột nhiên biến mất khỏi giao diện, tạo cảm giác mất mát dữ liệu.

1. Trực quan theo thao tác của người dùng: What you see is what you get (WYSIWYG)

**VD:** Giúp người dùng xác định được vị trí, tiến trình trong nghiệp vụ đang thao tác (đã làm những gì, còn phải làm gì tiếp?)

1. Cho phép người dùng có nhiều phương án thao tác trên giao diện

**VD của Tú:**

Với công việc tìm kiếm 1 đoạn văn bản

Đối với người dùng thành thạo: Sử dụng hotkey[Ctrl + F]

Đối với người mới sử dụng: Menu ⇒ Home ⇒ Find

**Ví dụ của Phong:**

Với việc gửi một hình ảnh

Đối với người dùng thành thạo: kéo và thả ảnh muốn gửi vào

Đối với người dùng mới: chọn nút “gửi hình ảnh”

1. Tránh lãng phí, dư thừa không gian giao diện

**VD của Tú:** Không dành ra cả 1 gốc diện tích chỉ để hiện cứng các thông báo hướng dẫn, hay thông báo lỗi, nên cho người dùng ẩn hiện theo ý muốn.

**Ví dụ của Phong:** chỉ hiện thông báo lần đầu mở ứng dụng. Sau này, người dùng có thể tự vào mục lục để tìm kiếm các trợ giúp

1. Bố trí các thành phần trên giao diện sao cho những thao tác quan trọng thường dùng đặt tại các vị trí dễ nhìn, dễ quan sát gây chú ý cho người dùng. Các thao tác có rủi ro cao, nguy hiểm dữ liệu, đòi hỏi tính chuyên môn cao thì bố trị vị trí thích hợp, hạn chế sai sót, rủi ro.

**VD của Tú:**

Nút thoát chương trình nằm ở vị trí góc trên bên phải, khó thao tác nhầm

Các nút hủy thao tác, xóa dữ liệu sẽ được làm ít nổi bật, vị trí khó thấy hơn các nút đặt mua, xác nhận,…

**Ví dụ của Phong:**

Các chức năng thường sử dụng nên được nhóm lại vào một tùy chọn. Khi người dùng di chuột vào sẽ hiện ra các chức năng hay dùng

Hiện thông báo xác nhận hành động đối với các thao tác quan trọng gây thay đổi dữ liệu

1. Nhất quán về dạng các giao diện và thành phần trên giao diện

**VD:**

Đỏ: lỗi

Cam: Cảnh báo

Xanh: Thông báo

Vị trí nút lệnh, các hotkey ở nhiều form khác nhau phải tương đồng, nhất quán với nhau.

# Câu 11:

a. Khi thiết kế hệ thống phần mềm, cần lấy hệ thống giao diện làm trung tâm, vì sao?

b. Phát biểu trên có mâu thuẫn gì với phát biểu “Khi xây dựng phần mềm phải lấy người sử dụng làm trung tâm không”? Giải thích? (Vì sao khi thiết kế hệ thống phần mềm cần quan tâm đến giao diện)

**11b:** Không mâu thuẫn, vì người dùng tương tác với phần mềm thông qua giao diện. Tất cả các thiết kế khác đều phải xuất phát từ giao diện, hay nói cách khác, người dùng và giao diện là một.

**11a:** Bởi vì tất cả các phần mềm được sản xuất ra đều là để phục vụ cho người dùng, người dùng tương tác với phần mềm thông qua giao diện, vì vậy mọi thiết kế khác (xử lý, bảo mật, dữ liệu,…) đều phải xuất phát từ giao diện. Thêm vào đó, việc trao đổi để xây dựng ý tưởng, khảo sát, phân tích hệ thống dựa trên cơ sở phác thảo giao diện trước.

# Câu 12:

Để đánh giá chất lượng của hệ thống giao diện, thường phải đảm bảo những nguyên tắc gì? Căn cứ những nhóm tiêu chí nào? Minh họa thực tế?

***Nguyên tắc***

1. Lấy người sử dụng làm trung tâm
2. Xem xét trên 2 phương diện: sản xuất (kĩ thuật) và sử dụng (hiệu quả)
3. Không có bộ tiêu chí nào là chung cho tất cả giao diện

**VD:**

Cùng 1 phần mềm nhưng tiêu chí đánh giá có thể khác nhau tùy vào: dạng, loại phần mềm, mục đích sử dụng, đối tượng người dùng, người đầu tư.

Bộ tiêu chí này còn có thể thay đổi theo thời gian, tùy thuộc vào yếu tố công nghệ kĩ thuật phát triển, như lúc trước khi phát triển phần mềm thì thường yêu cầu cao về tiết kiệm bộ nhớ.

1. Tất cả bộ tiêu chí cần được xác định trước khi xây dựng phần mềm: xác định giới hạn, phạm vi và tiêu chí đánh giá, nếu không thì phần mềm sẽ không có điểm dừng, và phải giao kèo trước các giới hạn này với tổ chức thụ hưởng.
2. Ngoài các tiêu chí đương nhiên (chạy đúng, chạy đủ,…) ta có các nhóm tiêu chí gợi ý như sau khi đánh giá màn hình giao diện:

* **Tính đúng đắn:** so với phân tích, thiết kế hệ thống phần mềm và các yêu cầu cụ thể đã đặt ra.
* **Tính tiện dụng:** sử dụng dễ nhất, người dùng ít tốn thời gian để học cách dùng. Người dùng có khả năng khai thác sử dụng hết tất cả các tính năng của phần mềm có trên gd
* **Tính hiệu quả:**

Ít tốn công sức, chi phí, thời gian khi thiết kế giao diện.

Khả năng sinh lời cao.

Đơn giản: số lượng màn hình ít , thành phần mh ko phức tạp.

Người dùng ít thao tác, không lặp những thao tác giống nhau, hạn chế sai sót khi thao tác trên giao diện.

Giao diện được thiết kế sao cho việc sản xuất phần mềm ít tốn nhất, do thiết kế giao diện ảnh hưởng đến vấn đề lập trình logic phía sau.

*Ngoài ra còn có các tiêu chí khác, tùy dạng phần mềm:*

* **Bảo mật (Security):** tránh truy xuất thông tin trên phần mềm bất hợp pháp (Hacking)
* **An toàn (Safety)** thao tác: tránh hư hỏng, mất thông tin, dữ liệu (Virus, Backup/ Restore)

# Câu 13:

a. Giới thiệu một số dạng màn hình giao diện thông dụng (tên gọi, ý nghĩa sử dụng, nội dung)?

b. Một màn hình giao diện có thể thiết kế theo nhiều trong số các dạng nêu trên đc ko? Vì sao? Minh họa thực tế?

**13a:** Một số màn hình giao diện (truyền thống) thông dụng:

|  |  |  |
| --- | --- | --- |
| **Loại giao diện** | **Ý nghĩa sử dụng** | **Nội dung chính** |
| Màn hình chính  *Winform: Main Form*  *Webform: Home Page* | Chọn công việc, chọn form | Danh sách các công việc |
| Form Nhập dữ liệu lưu trữ | Lưu trữ thông tin mới | Form nhập dữ liệu: danh sách các thông tin phải nhập  Control lưu trữ |
| Form Nhập dữ liệu xử lý | Nhập thông tin cho phép thực hiện 1 thao tác khác (có thể chuyển điều khiển sang form khác) | - Danh sách thông tin nhập (input)  - Đăng ký chuyển hoặc xử lỳ |
| Màn hình kết xuất kết quả (Report) | Trình bày kết quả cho người dùng (Report) | Danh sách các thông tin kết quả cần kết xuất |
| Màn hình thông báo trạng thái | Thông báo Trạng thái hiện tại, cảnh bảo, nhắc người dùng thực hiện công việc | Danh sách nội dung trên hình tượng cần thông tin đến người dùng |
| Màn hình tra cứu, tìm kiếm  (Search, Find, Locate) | Nhập thông tin cơ bản, tra cứu các thông tin khác đã lưu trữ  Chú ý: Cá biệt nhập dữ liệu xử lý | - Nhập các thông tincơ bản  - Thông tin kết quả tra cứu (danh sách) |

**13b:** Một màn hình có thể thiết kế theo nhiều dạng nêu trên, do xu hướng hiện tại hướng tới form giao diện mở, đặt biệt kết hợp AI (Machine learning). Nghĩa là form động, tự thay đổi theo thực tiễn sử dụng thậm chí là tùy ý thêm bớt theo ý của người dùng.

**VD:** Nhà thông minh (IoT) tự động nhận diện thói quen sử dụng đồ vật của người dùng để tự động hóa một số tác vụ như mở kênh truyền hình yêu thích, mở bài nhạc hay nghe,…

# Câu 14 (Thiếu VD):

So sánh, phân biệt những điểm khác nhau, giống nhau cơ bản giữa các loại giao diện: Winform, Webform, Mobileform? khi thiết kế các gd nêu trên, cần quan tâm những yếu tố đặc thù nào? Minh họa.

Winform:

* Không gian giao diện rộng rãi ổn định, rộng rãi hơn so với mobile.
* Có chức năng thoát chương trình (this.close(), Application.Exit(),..).
* Có MessageBox
* Sử dụng scrollbar để tải thêm dữ liệu
* Chỉ phù hợp với văn phòng, nhà cố định đã cài win app
* Thường thêm sửa bằng form (textbox, label,…)

**VD: ?**

Webform

* Không gian rộng rãi hơn so với mobile, tuy nhiên còn phụ thuộc nhiều vào loại trình duyệt, độ phân giải ⇒ Khắc phục bằng table và thường có 2 phiên bản: PC và Pocket (Dành cho mobile)
* Đa dạng giao diện, phù hợp với nhiều dạng dữ liệu khác nhau
* Cơ động, không cần cải đặt ⇒ quy mô toàn cầu, cần phải quan tâm đến văn hóa, ngôn ngữ, phong tục tập quán của nhiều nước
* Không có chức năng thoát như winform, thay vào đó sử dụng đăng nhập/đăng xuất
* Không có MessageBox riêng, các phương án thay thế: nhờ MessageBox của trình duyệt, hiện thông tin thẳng trên nền trang web, sử dụng modal (HTML, CSS, JS)
* Sử dụng phân trang để tải dữ liệu mới
* Thường chỉnh sửa dữ liệu ngay trên lưới

**VD: ?**

Mobileform

* Thoát: dùng cảm biến, cũng có thể thoát hẳn chương trình giống như Winform
* Sử dụng scrollbar để tải thêm dữ liệu
* Không gian giao diện hạn chế, phải cân nhắc tránh lãng phí, lạm dụng khi thiết kế giao diện
* Người dùng cơ động vì điện thoại có thể đem theo bên người bất cứ đâu, không như winform dùng PC ⇒ Phù hợp với người dùng thường xuyên di chuyển như shipper
* Có MessageBox riêng như Winform

**VD: ?**

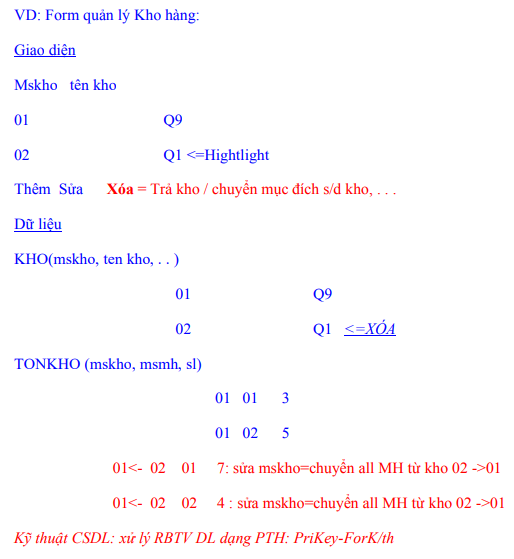
# Câu 15:

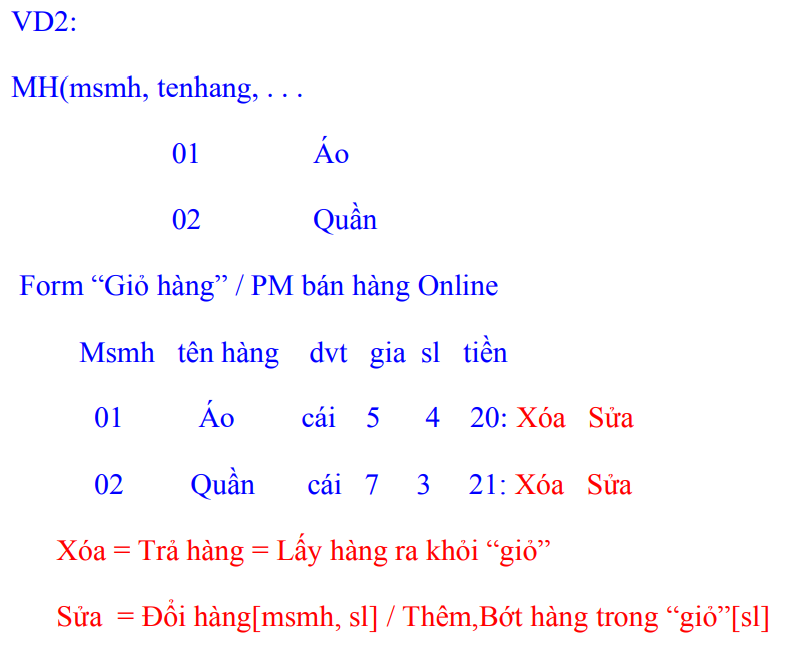
Thiết giao diện ảnh hưởng như thế nào đến tâm lý người dùng? Minh họa thực tế?

(Vì sao khi thiết đánh giá phần mềm cần quan tâm tâm lý người dùng)

Thiết kế hệ thống giao diện phụ thuộc trực tiếp vào yếu tố tâm lý người dùng, đặc biệt là thiết kế trợ giúp. Vì vậy khi thiết kế giao diện (và trợ giúp), cần quan tâm những yếu tố tâm lý sau đây của người dùng:

1. Người dùng không thành thạo về công nghệ phần mềm và thậm chí thao tác máy tính ko tốt. Vì vậy, khi thiết kế giao diện phần mềm,cần đóng vai người dùng không thành thạo công nghệ phần mềm

**VD1:** Xóa kho, sửa kho ⇒ Trả kho/chuyển mục đích sử dụng kho, sửa thông tin kho

**VD2:** Xóa, sửa ⇒ Trả hàng, Đổi hàng

1. Khi người dùng gặp phải 1 sai sót, lỗi do thao tác, người ta thường liên tưởng đến toàn bộ hệ thống phần mềm, đánh giả chất lượng pm kém dẫn đến tâm lý ngại dùng phần mềm, từ chối sử dụng.

**VD:** Xét bài tập đánh giá giao diện quản lý dự án: Nếu cho người dùng sửa mã số nhóm dự án ngay trên Form quản lý dự án thì sẽ làm cho tên dự án biến mất khỏi danh sáchdự án dẫn đến tâm lý người dùng lo lắng,có sự mất mát dữ liệu, trong khi thực tế dữ liệu không mất, mà chỉ chuyển sang nhóm khác, nên ko còn trong danh sách nhóm này nữa. Xét về kỹ thuật thiết kế giao diện không sai, nhưng xét vè tâm lý người dùng thì không đạt yêu cầu khiến người dùng không còn tin tưởng phần mềm dẫn đến ngại sử dụng.

1. Người dùng ngại đọc chữ, thay vào đó họ thường quan sát hình

**VD của Tú:** Nên chèn thêm một số hình ảnh (icon) trước mỗi chữ của hành động trong button, menu,… để thêm sinh động, trực quan, bỏ qua được phần nào rào cản ngôn ngữ. Hoặc với các hành động đơn giản, dễ hình dung có thể bỏ hẳn phần chữ chú thích. Tuy nhiên cũng không nên quá lạm dụng sẽ dẫn đến tác dụng ngược, gây khó hiểu.

1. Có thể thay đổi, cập nhật giao diện theo thời gian, tránh “nhàm chán” cho người dùng sử dụng lâu dài. Tuy nhiên, không nên làm quá “bất ngờ”, thay đổi đột ngột sẽ tạo tâm lý e ngại.
2. Hình thành sự tin tưởng của người dùng đối với sản phẩm phần mềm (tương lai), trước khi bàn giao phần mềm, thông qua các thủ tục về tâm lý.

**VD:** giới thiệu trước về tính hiệu quả, tiện dụng,… để hình thành niềm tin với người dùng, qua đó họ sẽ dễ thông cảm và chia sẻ, phối hợp với bên sản xuất phần mềm có phát sinh sự cố kỹ thuật trong tương lai.

# Câu 16:

Trình tự các bước thiết kế 1 form giao diện trên hệ thống phần mềm? Minh họa thực tế?

**B1:** Xác định vị trí các form trong tổng thể kiến trúc phần mềm

**B2:** Xác định loại form: winform, webform, mobileform, do nhà sản xuất phần mềm đề nghị trên cơ sở kết hợp nhu cầu và điều kiện người dùng, dựa vào kiến trúc pm đã thiết kế. Loại nào ưu tiên hơn? Cần bảo mật không? Có giới hạn quyền không? Có cần tách dùng form phụ không?

**VD:** Tùy thuộc vào từng nghiệp vụ thực tế mà linh hoạt chọn giữa 2 phương án cho nghiệp vụ đó.Sử dụng form phụ với các tác vụ cần độ chính xác cao, số lượng dòng dữ liệu cần nhập ít, và không cần tách form phụ với các trường hợp ngược lại.

**B3:** Xác định mục đích chính của form là để làm gì?

**B4:** Xác định (các) đối tượng sử dụng là (những) ai dùng? Các quyền truy xuất tương ứng từng đối tượng sử dụng (cấp độ bảo mật)?

**VD của Tú:** Xét form nhập điểm theo lớp của 1 phần mềm quản lý đào tạo. Với quyền truy cập là thư ký khoa thì người dùng đó chỉ được phép xem, kiểm dò lại điểm với những lớp thuộc khoa mình phụ trách, còn với quyền truy cập là cán bộ phòng khảo thí thì có thể kiểm dò, xem điểm cho tất cả lớp thuộc tất cả khoa

**Ví dụ của Phong:**

Xét trang quản lý của một website:

Một tài khoản quản trị viên sẽ được truy cập vào nhiều chức năng hơn ( cài đặt, quản lý user)

Một tài khoản cộng tác viên sẽ chỉ được truy cập các tính năng liên quan tới vai trò của mình( them – sửa – xóa bài viết)

**B5:** Thiết kế form, dựa trên 8 nội dung thiết kế phần mềm

**B5.1**: Thiết kế sơ đồ tĩnh của hệ thống forms và xác định vị trí của chúng trong sơ đồ đó. Mô tả hệ thống tĩnh của giao diện thường sử mô hình tree (đồ thị).

* Mô tả tĩnh là logic khi thiết kế, chưa quan tâm vấn đề chuyển điều khiển giữa các form
* Chỉ có tên mô hình giao diện (và có thể có chỉ số nhận diện: 1. 1.1. 1.2. )

**VD:** WebApp: SiteMapPath

**B5.2**: Phác thảo hình thức, vị trí các thành phần trên form (kể cả các nút lệnh đều khiến), màu sắc,...). Lần lượt thiết kế từng form gd trong mối quan hệ giữa các form giao diện theo sơ đồ cây nêu trên.

**B5.3**:

*Kịch bản khai thác:* DFD tĩnh gọn hoặc Activity Diagram/UML, thiết kế các sơ đồ kịch bản khai thác sử dụng các form giao diện.

* Thường dùng sơ đồ DFD tĩnh gọn (không có kho dữ liệu)
* Mô tả trường hợp người dùng thao tác và làm việc đúng các nghiệp vụ trên form

**VD của Tú:** Khi người dùng chọn “thêm dự án” khi đang sửa dang dở 1 dự án chưa lưu ⇒ DFD không thể phát hiện được lỗi này của người dùng.

**Ví dụ của Phong**: Khi người dung đang sửa một dự án thì ta sẽ vô hiệu hóa chức năng them, xóa,…

Kết hợp sơ đồ chuyển trạng thái Petri

* Trong form giao diện (thay đổi trạng thái form): Thêm ⇒người dùng nhập ..
* Chuyển sang form giao diện khác:

**VD:** Tại form giao diện quản lý hàng hóa, người dùng bấm nút thoát ⇒ trở về màn hình chính

**VD của Tú:** Khi người dùng chọn “thêm dự án” khi đang sửa dang dở 1 dự án chưa lưu.

⇒ Sử dụng Petri để thể hiện được lỗi này của người dùng ⇒ Vô hiệu hóa nút “thêm dự án”

*Kịch bản xử lý phát sinh lỗi:* sử dụng Petri, tuy nhiên, chỉ tìm những trường hợp cơ bản, không thể tìm hết được

**VD:** Mô tả trường hợp người dùng thao tác và làm việc sai nghiệp vụ như nhập thiếu dữ liệu bắt buộc trên form.

**B6:** Vấn đề bảo mật, quyền tương ứng với từng đối tượng ⇒ quản lý bằng các thuộc tính Enable, Visible cho cả form hoặc các control cụ thể.

**VD của Tú:** Vai trò giám sát dự án thì không cho thêm xóa sửa dự án. Vai trò tài chính kế toán thì chỉ được thay đổi chủ đầu tư.

**Ví dụ của Phong**: Vai trò giám sát thì chỉ quản lý dự án. Vai trò kế toàn thì sẽ không truy cập được vào các chức năng của chủ dự án

**B7:** Thiết kế trợ giúp cụ thể cho từng form và trợ giúp cho liên form (trợ giúp chung)

**Lưu ý:** Trợ giúp nên tách ra thành 1 cửa sổ riêng, có khả năng ẩn hiện để không chiếm chỗ vô ích và gây khó chịu cho người dùng.

**B8:** Thiết kế cơ chế và thành phần đóng gói, cài đặt, chuyển giao: tự động [font, kết nối db,...], bản quyền, hỗ trợ người sử dụng có bản quyền,…

**B9:** Thiết kế các case kiểm thử nếu cần, tung ra các bản beta. Ngoài ra, có thể phải dự kiến về khả năng mở rộng phần mềm trong tương lai, do nhu cầu của người dùng hoặc sự thay đổi, phát triển của nền tảng khoa học công nghệ

# Câu 17:

Các cấp độ thiết kế hệ thống form giao diện của phần mềm? Trình tự thực hiện trong thực tế như thế nào? Minh họa thực tế

Các cấp độ thiết kế form trên hệ thống phần mềm: có 3 cấp độ

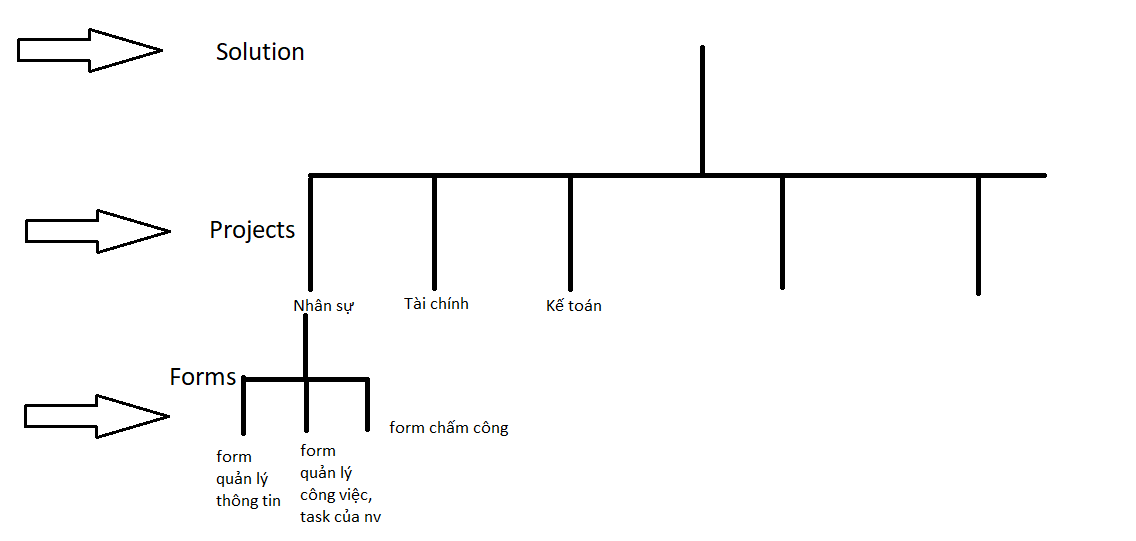
Cấp 1: từng form (9 bước câu 16)

Cấp 2 - Project: module (components) của từng nhóm chức năng

Cấp 3 - Solution: Tất cả hệ thống phần mềm

Thực tế thực hiện: Cấp 3 ⇒ Cấp 2 ⇒ Cấp 1

**VD của Tú:**



# Câu 18:

Giới thiệu một số hình thức và dạng kết cấu mục chọn thực đơn thông dụng hiện nay trong các hệ thống phần mềm? Các Nguyên tắc thiết kế thực đơn phần mềm? Minh họa thực tế?

Có nhiều hình thức, thông dụng nhất hiện nay là Dropdown menu và Ribbon menu

Một số ứng dụng sử dụng Dropdown menu: Diagrams.net, Google docs, Visual Studio, VNExpress.net,…

Một số ứng dụng sử dụng Ribbon menu: Mirosoft office (Word, Excel, PowerPoint,…), nld.com.vn, tuoitre.vn,…

Có nhiều dạng kết cấu thông dụng nhất hiện nay là kết giữa hướng đối tượng (Object Oriented – OO) với hướng chức năng (Function Oriented – FO)

**Thiết kế kết cấu các mục chọn Menu**

**OO:**

**B1**: Xác định danh sách các đối tượng quản lý của hệ thống phần mềm.

**Ví dụ của Tú:** Quản lý mua bán bánh kem: Sản phẩm, đơn hàng, tài khoản, thông tin cá nhân

**Ví dụ của Phong**: Quản lý nhà hang: nhân viên, đơn hàng, nguyên vật liệu, cơ sở vật chất, tài chính

**B2:** Nhận diện các đối tượng trung tâm quản lý của phần mềm

**VD của Tú:** Sản phẩm, Đơn hàng, tài khoản, thông tin cá nhân.

**Ví dụ của Phong:** Đơn hàng, nhân viên, nguyên vật liệu.

**B3**: Xác định người sử dụng của menu (Nhânviên quản lý, kế toán, khách hàng)

và quyền truy xuất tương ứng

**B4:** Thiết kế mục chính của menu dựa theo đối tượng trung tâm (B2) và quyền truy xuất (B3)

**B5:** Thiết kế các mục chọn của từng thực đơn (MenuItem, Ribbon)

**FO:**

**B1:** xác định danh sách các chức năng của hệ thống phần mềm:

**VD của Tú**: Quản lý mua bán bánh kem: Mua hàng online, duyệt đơn hàng, thống kê, quản lý thông tin,…

**Ví dụ của Phong:** Quản lý nhà hàng: đặt món online, duyện trạng thái đơn hàng, quản lý thông tin khách hàng

**B2:** Nhận diện các chức năng trung tâm quản lý của phần mềm

**VD của Tú:** Mua hàng, duyệt đơn hàng

**B3:** Xác định người sử dụng menu: Nhân viên kế toán, nhân viên quản lý, khách hàng,… và quyền truy xuất tương ứng

**B4:** Thiết kế mục chính của menu bằng chức năng trung tâm (B2) và quyền sử dụng (B3)

**B5:** Thiết kế các mục chọn của từng thực đơn (MenuItem, Ribbon)

**Nguyên tắc thiết kế thực đơn của hệ thống phần mềm**

1. Các mục chọn của thực đơn phải có đầy đủ tất cả chức năng phần mềm và công cụ cho người dùng
2. Gợi nhớ và phù hợp thực tế sử dụng phần mềm, phân chia nhóm thực đơn và đặt tên các mục chọn thực đơn giúp người dùng nhớ đc các mục họ cần chọn và dùng các thuật ngữ đúng chuyên môn, phổ thông (hạn chế thuật ngữ địa phương)
3. Kết cấu thực đơn càng đơn giản càng tốt; hạn chế trùng lắp, trừ 1 số trường hợp do chức năng sử dụng có liên quan đến nhiều nhóm thực đơn
4. Thiết kế kết cấu thực đơn ưu tiên theo người dùng chưa thành thạo (do người dùng thành thạo thường xài hotkey)
5. Linh hoạt, có nhiều tùy biến và thay đối theo từng đối tượng người dùng
6. Không chiếm nhiều không gian nhưng phải tập trung đc chú ý của người dùng
7. Tùy từng trường hợp đặc thù phần mềm mà sẽ có những nguyên tắc khác

# Câu 19 (13b + 14):

Giới thiệu một số đa dạng hóa hình thức trình bày các form? Minh họa thực tế

Cơ chế form thay đổi tùy vào tình huống và điều kiện sử dụng của người dùng

Đặc biệt là Webform, vì sd toàn thế giới với tất cả đối tượng người dùng , không cần biết người dùng là ai. Vì vậy, khi thiết kế các form phải thực hiện đa dạng theo ngôn ngữ, thói quen, phòng tục, tập quán, mục tiêu, luật pháp,… **(14)**

Khuynh hướng tương lai là form tự tùy biến & đa dạng dựa vào AI nghĩa là form có khả năng tự biết nhu cầu người dùng và điều chỉnh theo. Tuy vậy, có thể đôi lúc hoạt động không chính xác nên vẫn cho người dùng chủ động chọn thủ công **(13b)**

Công cụ hỗ trợ thiết kế và lập trình form như trên hiện nay chưa đáp ứng, trong thời gian tới sẽ có hỗ trợ.

**VD:** Form biến đổi ngôn ngữ theo mong muốn của người dùng, triển khai bằng cách lưu các label, các text tĩnh dưới dạng các key, tương ứng với mỗi ngôn ngữ sẽ có 1 file, mỗi key đó trong từng file sẽ có các value khác nhau, tương ứng với các ngôn ngữ khác nhau.

# Câu 20:

Tối ưu hóa form giao diện là gì (mục đích , tiêu chí)? Tách/ghép các form gd phải đảm bảo những nguyên tắc gì? Minh họa thực tế?

Tối ưu hóa các form chủ yếu dựa trên cơ sở **(tiêu chí)** tách và ghép các form. Việc tách và ghép các form giao diện sẽ ảnh hưởng đến các thiết kế khác như lập trình, trợ giúp, bảo mật, cơ sở dữ liệu, kiểm thử,… Vậy nên **mục tiêu, mục đích** của việc tách ghép, tối ưu hóa các form là hướng tới sự đơn giản.

**VD:** Xét trong phần mềm quản lý dự án, nếu thêm dự án mới sử dụng form phụ sẽ hạn chế được kịch bản phát sinh lỗi (không thể bấm nút thêm, xóa, sửa) nhưng nhược điểm là số lượng form phải thiết kế lớn, thao tác xử lý chậm.

# Câu 21:

Hãy cho 1 ví dụ về việc tách/ghép form/page có ảnh hưởng đến mã lệnh (code)

Việc tách / ghép form sẽ ảnh hưởng các thiết kế khác như code, trợ giúp, bảo mật, giao tiếp khác, database, kiểm thử, cài đặt,… Mục đích cuối cùng của việc này là hướng tới tính hiệu quả (sự đơn giản) trong thiết kế.

**VD:** Xét chức năng thêm dự án của phần mềm quản lý dự án (BT5 / Trang 5)

1. **Phương án đổi nhãn:**

Phương pháp: Đổi chữ của button “Thêm” thành “Lưu”

Điều này sẽ dẫn đến;

* **Codes** phức tạp :

if (nhận diện: thêm…)

{ …chuẩn bị thêm…}

else (nhận diện:lưu…)

{ ….lưu d/a mới vào db…}

* **Trợ giúp, bảo mật:** phải thiết kế cho 2 trường hợp thêm và lưu, đè chồng lên nhau ⇒ phức tạp
* **Phải tránh trường hợp người dùng thao tác vô lý**. Ví dụ đang trong trạng thái “Thêm” (Nhập liệu thông tin để thêm mới) thì bấm nút “Xóa”, “Sửa”. Giải pháp là sử dụng thuộc tính Enable/Visible để vô hiệu các nút bị cấm.

1. **Phương án tách form riêng**

Phương pháp: Thêm form phụ (tách form…) thêm dự án

Điều này sẽ dẫn đến

* Tăng số lượng form, nhưng các thiết kế khác đơn giản hơn
* Codes đơn giản: form chính: {yêu cầu thêm dự án…}

{Gọi form phụ thêm dự án…}

form phụ:

{Nhập liệu…}

{Yêu cầu lưu…}

{Lưu dự án mới vào database…}

* Trợ giúp, bảo mật
* Người dùng tập trung, gần như không nhầm
* Tránh trường hợp truy xuất không hợp lệ rồi lại phải đi chặn
* Kiểm thử bằng phương pháp hộp trắng dễ hơn

Tiêu chí: Xem lại câu 12

Tiêu chí khác nhau : người dùng, tốc độ, bộ nhớ, …

**Chú ý của Tú: người dùng thực tế thông qua k/s người dùng free [test, beta] ⇒ ??? k/s là gì ☹ Không biết dịch sao…**

Có nhiều phương pháp để tối ưu hóa giao diện, tách/ghép form là 1 trong những phương pháp đó, ngoài ra còn có: tái cấu trúc các thành phần trên gd, . . .

Tách form giao diện có nhiều dạng: bổ sung các form phụ, liên kết hỗ trợ form, hoặc **or sd xml (hoặc sử dụng xml???)**

**VD**: ….

Việc tách/ghép form sẽ ảnh hưởng codes nhưng không ảnh hưởng database

**VD:** form quản lý dự án (BT5 / Trang 5): thêm bằng form phụ

# Câu 22:

Trình tự thực hiện việc thiết kế hệ thống form giao diện theo theo hệ thống database đã có? Minh họa?

HƯỚNG DẪN TRÌNH TỰ THỰC HIỆN BÀI LÀM (tham khảo)

**B1:** Phân tích và xác định tính hợp lý của database: chủ yếu là vấn đề chuẩn hóa 3NF (*Các “quan hệ” thuộc tính không khóa phụ thuộc hàm đầy đủ và không bắt cầu vào tập khóa chính, đơn trị*), nếu chưa chuẩn hóa thì chuẩn hóa.

**B2:** Xác định khả năng mở rộng database trong thực tế (Chú ý: phải xác định phạm vi db: lộ trình)

**VD:**

**Khả năng mở rộng**

* Do MSVN (Mã số nhân viên?) trong ddh (Đơn đặt hàng?) ⇒ Table nv (nhân viên?)
* Điểm trong KH (Khách hàng?) ⇒ cơ chế cập nhật diem như thế nào
* Do có giao hàng ⇒ khu vực & chuyên môn giao hàng
* Hàng phải lưu kho và tonkho
* Vân vân,…

**Phạm vi DB (cơ sở dữ liệu ??)**

* NV(Nhân viên) ⇒ Tiền lương ⇒ Phần mềm kế toán
* Tài sản cơ sở vật chất ⇒ Phần mềm quản lý tài sản
* NV(Nhân viên) ⇒ Phần mềm quản lý nhân sự
* Vân vân,…

⇒ Thiết kế thành phần giao tiếp khác: thiết bị ngoại vi, các hệ thống phần mềm khác: phần mềm kế toán, quản lý nhân sự, quản lý tài sản, phần mềm quản lý marketing, . .

**B3:** Xác định các đối tượng sử dụng liên quan

* Khách hàng, nhân viên quản lý hàng hóa, trưởng bộ phận giao hàng, nv giao hàng, thủ trưởng đơn vị, . . .⇒ quyền liên quan từng đối tượng sử dụng.
* Xác định phần mềm phải trong tổng thể của 1 kiến trúc phần mềm ⇒ do đó, tùy trường hợp phải phác thảo kiến trúc của hệ thống phần mềm liên quan: định phần trích pm đang thiết kế là ở đâu trong tổng thể kiến trúc pm đó (xác định phần mềm hiện tại giao tiếp với các hệ thống ngoài nào: phần mềm kế toán, phần mềm marketing,..)

**B4:** Xác định số lượng form và loại form:

* Thông thường số lượng form tối thiểu bằng số lượng đối tượng sử dụng

**B5:** Xác định các thông tin liên quan mỗi form:

* Đọc dữ liệu từ các table nào ?
* Ghi dữ liệu xuống các table nào ?
* Thông tin trên form là gì :
  + Input (nếu có) trên form, output (nếu có)
  + Điều khiển (control) trong form / chuyển form
* **Chú ý:** đảm bảo tất cả các dữ liệu trong db phải được dùng hết.

**B6:** vẽ phác thảo form (tham khảo 7 bước tk form; nhưng bỏ qua các bước chưa quan tâm)

Với mỗi form cần xác định rõ những nội dung sau trước khi vẽ phác thảo

1. Xác định đối tượng (các đối tượng) sử dụng: ai (những ai) dùng ? và các quyền truy xuất tương ứng từng đối tượng sử dụng (cấp độ bảo mật)?

**VD:** khách hàng

1. Mục đích chính của form : để làm gì ?

**VD:** chọn hàng, đặt hàng, mua hàng

1. Loại (các loại) cần cùng : winform, webform, phoneform?

Loại nào ưu tiên hơn? cần bảo mật không? có giới hạn quyền không? tách (dùng form phụ) / ghép form ....

1. Các table đọc dữ liệu lên? ghi dữ liệu xuống : . . . .?

Input, trên form, output ?

1. Các xử lý chính trong form, chuyển form? “kịch bản khai thác”: đúng, phát sinh lỗi,...
2. Màu sắc, hình dạng, xử lý đặc thù . . . .

# Câu 23:

Một số yêu cầu và nguyên tắc cơ bản về thiết kế **thành phần cơ sở dữ liệu** của hệ thống phần mềm

***Nguyên tắc:***

Phù hợp, nghĩa là không dư thừa, dự kiến mở rông, không thiếu, không lạc hậu,…

**Chú ý:** “Không lạc hậu” có ý nghĩa là hầu hết DB của App hiện nay đều triển khai trên Internet (cloud) ⇒ Phân tán, DWH[Dim, Fact], DMN[AI], BigData, …

***Yêu cầu***

* Ít nhất 3NF = đơn trị, đầy đủ, trực tiếp
* DB luôn cho phép mở rộng trong tương lai
* DB linh hoạt
* DB dễ thích nghi
* Khuynh hướng DB phi cấu trúc & vài trò XML
* Phân tán & triển khai sử dụng trên các nền tảng Internet
* Có xử lý Datawarehouse (nhà kho dl) và BigData (phù hợp IR4 = CM CN lần 4), khai phá

# Câu 24 (Thiếu VD):

Mối quan hệ giữa giao diện và dữ liệu?

**Tú: Câu này thầy ghi lú như con cú nên chịu khó xem file của thầy nha, ở đây tôi chỉ suy diễn bừa thôi**

Database tác động đến giao diện:

* Database luôn mở rộng ⇒ Giao diện phải thêm xóa chỉnh sửa cho phù hợp
* 1 database phù hợp nhiều dạng giao diện

Giao diện tác động đến database

* Xét tính hợp lý của giao diện trước ⇒ cải tiến database?
* Xét tính mở rộng của giao diện để thiết kế database

Thiết kế giao diện sẽ quyết định, ảnh hưởng đến việc thiết kế database và ngược lại:

* Nếu thiết kế database chặt chẽ, rõ ràng,…⇒ thiết kế giao diện sẽ rõ ràng, đơn giản, chuẩn và hợp lý
* Nếu thiết kế giao diện đơn giản, rõ, hiệu quả,…⇒ thiết kế database rõ và chuẩn,..

**VD: ?**

# Câu 25:

Các nội dung Thiết kế hệ thống CSDL cho 1 HTPM

1. Thiết kế DB nghiệp vụ, liên quan trực tiếp phần mềm

2. Thiết kế hệ thống CSDL bảo mật

Có 2 phương án:

* Thiết kế CSDL bảo mật riêng (tách biệt với CSDL nghiệp vụ)
* Thiết kế CSDL bảo mật chung với CSDL nghiệp vụ

Ràng buộc toàn vẹn = PTH:KCHINH - KHÓA NGOẠI (KHÓA ĐƠN - TỔ HỢP):

3. Thiết kế hệ thống CSDL dùng trên MobileApp (PhoneApp): Compact DB

4. Phân tán CSDL: xem lại Học phần CSDL phân tán

5. Vấn đề xử lý NULL trong CSDL

Người dùng bỏ qua nhiều thông tin khi nhập vào database (fields: allow null) ⇒ records quá nhiều null ⇒ chất lượng CSDL kém, ko có ý nghĩa sử dụng

Cách xử lý

* Lọc bỏ (rr)
* Trí tuệ nhân tạo (AI) ⇒ phỏng đoán các giá trị null
* Cho phép ngoại lệ, **VD:** khách hàng vãng lai

6. Thiết kế xử lý “khóa” trong CSDL

* Mở rộng, **VD: ?**
* Gợi nhớ: quản lý bằng xử lý PTH, **VD:** mã ID công dân
* Gọn (đơn giản) ⇒ dễ nhớ
* Vân vân,…

7. Thiết kế khả năng tương thích nhiều dạng dữ liệu cho phần mềm và vai trò XML file.

LT MetaData:

8. Thiết kế xử lý kho dữ liệu cho phần mềm

Khi thiết kế phần mềm có sử dụng CSDL thì theo thời gian, sẽ phát sinh lượng CSDL lưu lớn, có thể lên triệu triệu mẩu tin, dẫn đến các vấn đề trên phần mềm:

* Không truy xuất dc OR Chậm ⇒ vấn tin Data Warehouse, Integration
* Records quá cũ ko có trị sử dụng, cần phải xử lý backup hoặc delete, nếu để thì sẽ dẫn đến CSDL nặng, khai phá dữ liệu ra kết quả sai lệch so thực tiễn)
* Vân vân,…

⇒ Phải chú ý thiết kế xử lý Data Warehouse

Khai phá dữ liệu

Xử lý: khai khoáng, AI

# Câu 26 (Thầy bỏ trống + không giảng = Chưa nghĩ ra cách làm):

Nguyên tắc và các yêu thiết kế **thành phần** **xử lý** của hệ thống phần mềm? Minh họa

Thầy bỏ trống ☹

# Câu 27 (Thiếu VD):

a. Thế nào là thiết kế xử lý theo hướng chức năng: Top-Down

b. Thế nào là thiết kế xử lý theo hướng đối tượng: Bottom Up

Minh họa cho mỗi trường hợp

Gợi ý:

**27a: FO – Function Oriented (chức năng):** Từ chức năng tổng thể của PM ⇒ phân rã thành các chức năng nhỏ hơn ⇒ …⇒

Thao tác cụ thể: lập trình cài đặt xử lý

⇒ Top-Down

**27b: OO – Object Oriented (đối tượng) :** Xuất phát từ các đối tượng cụ thể liên quan đến PM: Lập trình xử lý theo hướng đối tượng (Class Object = Attributes & Operators || Behavior || Methods)…⇒ cấu thành phần mềm

⇒ Bottom-Up

# Câu 28 (Thiếu VD):

Thiết kế các hệ thống giao tiếp khác (giao diện) của hệ thống phần mềm

a. Giới thiệu một số dạng giao tiếp khác trong các hệ thống phần mềm? ví dụ minh họa?

b. Giới thiệu một số yêu cầu và nguyên tắc chung

**28a** Ví dụ một số giao tiếp ngoài: QR code, Bar code, thanh toán điện tử,…

**28b** Yêu cầu và nguyên tắc chung:

* Gắn liền với chức năng của phần mềm
* Đảm bảo không làm ảnh hưởng đến kết cấu phần mềm hiện có
* Đảm bảo an toàn và bảo mật đối với phần mềm
* Phù hợp với sự phát triển của khoa học công nghệ (không lạc hậu)
* Khái thác hiệu quả các chức năng của hệ thống

# Câu 29 (Thầy bỏ trống + không giảng = Chưa nghĩ ra cách làm):

Thiết kế hệ thống bảo mật và an toàn của hệ thống phần mềm

a. Thế nào là bảo mật HTPM? Minh họa?

b. Trong đơn vị sử dụng phần mềm: “thông thường, người có chức vụ càng cao thì quyền làm việc trên PM càng nhiều, người có chức vụ càng thấp thì quyền làm việc trên PM càng ít”; đúng hay sai? Vì sao? Thực tế thế nào?

c. Thế nào là an toàn HTPM? minh họa. Môi quan hệ giữa 2 vấn đề này

d. Giới thiệu một số yêu cầu và nguyên tắc cơ bản về hệ thống bảo mật của phần mềm? Minh họa?

# Câu 30 (Thiếu 30b)

a. Các cấp độ bảo mật trong giao dịch phần mềm trên Internet? Thường dùng chữ ký số trong các trường hợp phần mềm như thế nào?

b. Đề xuất cách nhận diện người dùng (thông qua các quyền) vd: nhận diện qua login,…

**30a:**

Các cấp độ bảo mật trong giao dịch phần mềm trên Internet:

* OS: đăng nhập HĐH dùng nội bộ
* Form (DB): dùng Table dữ liệu / DB : số người dùng hạn chế
* Passport hay chữ ký số: Số người dùng rất lớn, sử dụng Public key [PIN] & Private key dựa SHA / RSA (VD: user / facebook, gmail)

Chữ ký số:

* + bảo mật = tránh mạo danh, nặc danh
  + số người dùng quá lớn

**30b: Trống**

# Câu 31:

Phân quyền người sử dụng

a. Các cấp độ “quyền” thông dụng của người sử dụng?

b. Giới thiệu một số phương pháp nhận diện “quyền” của người sử dụng trên các phần mềm

Gọi ý:

**31a:**

Quyền:

* View
* Modify (Update)
* Delete
* Create
* Delete
* Copy
* Move
* Admin
* Owner (sở hữu)
* Liệt kê (List = Browser)

**31b:**

Nhận diện = chủ yếu qua Account đăng nhập vào PM;

**VD:** nhận diện NS?? phòng Nhân sự, Phòng Kế toán => phân cấp các quyền làm việc tương ứng với chức năng của cá nhân đó, NV NS không thực hiện các hạch toán kế toán, nhân viên kế toán không làm các NV NS

# Câu 32:

Thiết kế hệ thống trợ giúp (Help) của hệ thống phần mềm

a. Giới thiệu một số yêu cầu và nguyên tắc cơ bản về hệ thống trợ giúp của phần mềm?

b. Giới thiệu một số hình thức hỗ trợ người sử dụng thông dụng

Gợi ý:

**32a**

Một số yêu cầu và nguyên tắc cơ bản về hệ thống trợ giúp của phần mềm

1. Hạn chế người dùng sai sót khi sử dụng phần mềm, hỗ trợ tại mỗi bước đảm bảo hạn chế thao tác sai: ToolTip
2. Trực quan (What you see is what you get)

Vị trí trong tiến trình thao tác: Giới thiệu đồ họa

Giúp người dùng xác định được vị trí, tiến trình trong nghiệp vụ đang thao tác (đã làm những gì, còn phải làm gì tiếp?) ⇒ Hình dung tổng thể những việc họ sẽ làm trên PM

1. Trợ giúp phải triệt để
2. Tập trung các chức năng và thao tác chính (không lạm dụng)
3. Đảm bào người dùng khai thác sử dụng tất cả các chức năng mà phần mềm có

**32b**

* Text file
* Tooltip
* Đồ họa
* Sơ đồ cây
* Hướng dẫn context (bối cảnh)
* Làm giúp NSD theo mặc định (mặc định hóa)
* Vận dụng AI, nhận diện ý định của NSD để hỗ trợ (theo thói quen…)

# Câu 33 (chú ý KHÔNG PHẢI gợi ý làm bài):

Xét form màn hình của phần mềm quản lý dự án (trang 8 của đề cương) theo anh /chị những chức năng thao tác nào cần phải thiết kế hotkeys? Đề xuất hotkeys cụ thể.

Ưu tiên tk phím bấm tắt cho những tr/h nào? Vì sao? Minh họa

**Chú ý:**

1. **Không lạm dụng hotkeys**

* Chỉ thiết kế hotkeys cho những thao tác: thông dụng; chuyên môn/nghiệp vụ chính.
* Không nên thiết kế hotkeys cho các trường hợp sau:
  + Những thao tác, chức năng rủi ro cao, những thao tác chỉ dành cho những người am hiểu về phần mềm
  + Ít dùng

1. **Hotkeys nhất quán**

VD:

Đa số các sản phẩm phần mềm:

Hotkey thoát là [Alt+ F4]. Một số trường hợp dùng [Ctrl + C] (Cancel) không nên, vì nhầm với copy.

Hotkey copy / cut / paste là [Ctrl + C] / [Ctrl + X] / [Ctrl + V]

Hotkey undo / redo là [Ctrl + Z] / [Ctrl + Y]

Mục đích ta nên sử dụng các hotkey thông dụng này là để giúp người dùng dễ nhớ, không phải học nhiều, ít nhầm lẫn

1. **Hotkeys thống nhất ngôn ngữ giúp người dễ nhớ**

**VD:** Eng / Việt : [Ctrl + T] (tìm) / [Ctrl + F] (Find)

Không trình bày các phím/tổ hợp phím mặc nhiên

**VD:** mũi tên

# Câu 34:

Thiết kế các case kiểm thử của hệ thống phần mềm (Software Testing)

- Thế nào là kiểm thử phần mềm”

- Kiểm thử hộp đen, hộp trắng, hộp xám ?

- kiểm đơn vị, tích hợp, hệ thống, chấp nhận, kiểm tải, . . .?

- kiểm tĩnh, kiểm động?

- So sánh [PB] kiểm thử WinApp, WebApp, MobileApp/PhoneApp; . . . ?

- Giới thiệu một số yêu cầu và nguyên tắc cơ bản về kiểm thử phần mềm?

- Minh họa Thiết kế case kiểm thử hộp đen trong WinApp

- Minh họa Thiết kế case kiểm thử hộp đen trong WebApp

- Minh Thiết kế case kiểm thử hộp đen trong MobileApp/PhoneApp

- Minh họa Kiểm tải WebApp

# Câu 35:

- Thiết kế hệ thống cài đặt và chuyển giao của hệ thống phần mềm bao gồm những việc gì ? Minh họa?

- Giới thiệu một số yêu cầu và nguyên tắc cơ bản về hệ thống cài đặt và chuyển giao (Settup and Deployment) phần mềm

- So sánh [PB] thủ tục ĐÓNG GÓI VÀ THIẾT LẬP BỘ CÀI ĐẶT CÁC DẠNG PM KHÁC NHAU [WIN, WEB, MOBILE]

- Các nội dung Thiết kế hệ thống cài đặt của WinApp? Minh họa.

- Các nội dung Thiết kế hệ thống cài đặt của WebApp

- Các nội dung Thiết kế hệ thống cài đặt của MobileApp (PhoneApp)