

# KIẾN TRÚC và THIẾT KẾ PHẦN MỀM (Software Architecture & Software Design)

## Thi viết tự luận:

- Dạng chủ đề mở, khuyến khích sự sáng tạo (hợp lý, không lạc đề) của người học.
- Thời gian làm bài 90 phút.
- 2 đề phát xen kẽ
- Nội dung bài thi gồm 2 phần:

### [1] Kiến trúc phần mềm:

#### Bài tập:

#### DẠNG 1: CHO MÔ TẢ (HÌNH THỨC UĐ,...) OF HTPM; Y/C

- + XÁC ĐỊNH GIỚI HẠN HT PM
- + XD KIẾN TRÚC PHÙ HỢP ? GIẢI THÍCH LIÊN QUAN MÔ HÌNH?
- + XD DẠNG PM CHO MỖI THÀNH PHẦN / MÔ HÌNH KIẾN TRÚC ĐÃ TK Vd:

Tham khảo trên MS. Teams (Assignments)

. HTPM Quản lý đào tạo thông minh [Bài đã làm thêm]

. HTPM Quản lý thư viện thông minh [Bài đã làm thêm]

#### Câu hỏi: Tham khảo Bài 2 / Tr6 Đề cương

Giới thiệu một số chức năng hỗ trợ phân tích, xây dựng kiến trúc phần mềm trên Microsoft Visual Studio .NET (2010, 2012, 2013, 2015, 2017, 2019,...) hoặc một số công cụ tương đương khác? Giới thiệu một mô hình kiến trúc minh họa được thiết kế cho một phần mềm cụ thể (hoặc trích từ một phần mềm) trên các công cụ đó. [không giống nhau]: chấm điểm chỗ này

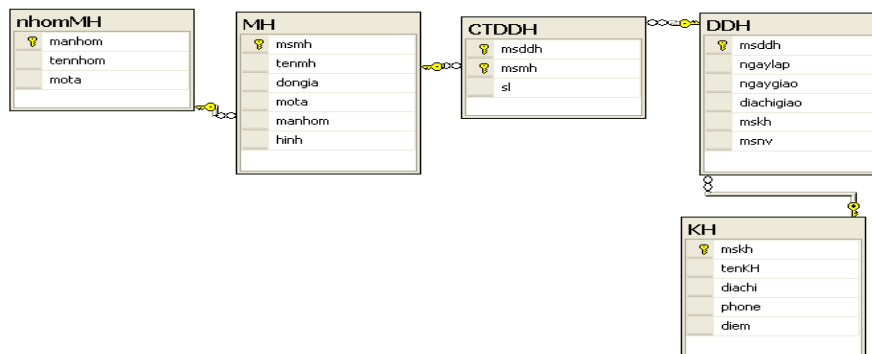
#### Gợi ý:

- Tham khảo bài tập 2 về nhà (bắt buộc)
- Biên tập lại ngắn gọn (cả giới thiệu công cụ + vd minh họa) đảm bảo sao cho có thể viết tay từ 15 – 20 phút
- Mỗi cá nhân tự làm minh họa riêng (để đảm bảo ko trùng với các bạn trong lớp)

### [2] Thiết kế phần mềm:

#### Bài tập: Dạng 1: THIẾT KẾ PM (GIAO DIỆN) PHẦN MỀM THEO MỘT DATABASE CHO TRƯỚC?

[Bài 6/ TR9 OR BTTL2/Tr6.ĐC]



2 đề là 2 DB ban đầu khác nhau

Câu hỏi: câu hỏi về các nguyên lý kỹ thuật & công nghệ trong thiết kế phần mềm.

Chọn 1 trong 35 câu trong danh mục = có thể kết cấu lại phù hợp thời lượng làm bài

#### Gợi ý:

- Biên tập lại ngắn gọn (trả lời + vd minh họa) đảm bảo sao cho có thể viết tay từ 15 – 20 phút
- Mỗi cá nhân tự làm minh họa riêng (để đảm bảo ko trùng với các bạn trong lớp)

#### 2 Bài tập:

- Biên tập lại ngắn gọn đảm bảo sao cho có thể viết tay từ 25 – 30 phút

### DANH MỤC CÂU HỎI:

#### Câu 1:

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

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.

HD:

=> Thiết kế KT PM, phù hợp thực trạng KT & CN

=> 1 số

1.1. Sử dụng phần mềm trên qui mô không giới hạn (toàn cầu): yêu cầu bảo mật cao.

. PM / Internet: -> nsd giao dịch / khác thác sd pm / all nơi, mọi lúc : ALL PM có k/n (khi tk KT PM có khai thác sd hay ko? )

. phát sinh :

[1] ng sx pm & ng s/d pm: ko nhất thiết biết nhau: thanh toán epayment

[2] Pm dễ bị Hacking = mất an toàn => bảo mật phải đủ mạnh & an toàn

[3] Pm s/d all cầu => số lượng người truy xuất / s.d rất lớn :

. kiểm tải (load test / webapp)

. số lượng nsd pm quá lớn => cks

VD WebMail => Số lượng account rất lớn (kho dl) : ko thể ktra đ nhập bằng DB (số record / db hàng tỷ => ko select...) ==> Chữ ký số = ktra đ n bằng chứng thư số; Nhờ giải thuật bảo mật, vd SHA/RSA

Chú ý: cks / quá account quá lớn / cấp passport

[4] Đa ngôn ngữ, đa văn hóa, đa tôn giáo, liên quan p luật nhiều q/ gia

[5] Bản quyền pm = mất an toàn hơn

[6] Mùi giờ = Client

1.2 Hệ thống máy tính mở: không giới hạn trong một PC :

+ <> Open Source:

+ PT HT PM dc tham gia bởi nhiều cá nhân / tổ chức: ko phụ thuộc k/g t/g ng ngữ

-> sở hữu trí tuệ = bản quyền / GPL

1.3 Phát triển công nghệ gắn liền với pháp lý

Sx & sd PM gắn liền với y/tổ p.lý

VD : FaceBook API, định vị nghe call => Quyền riêng tư (H of nhiều nước)

1.4 Giai đoạn nền CN 4.0: IoI, AI,...

\* Phù hợp với khuynh hướng KT & CN hiện tại và tương lai: Giới thiệu một số công nghệ liên quan:

+ Hyper-V (Virtualization) và các hệ thống mô phỏng (Emulator /Simulator)

Ảo hóa & mô phỏng => giảm rủi ro trong Software Testing

+ Công nghệ phòng & chống Hacker (Hacking)

g/d trên Internet

+ Phần mềm mã nguồn mở và GPL & GNU

Khai thác trí tuệ of all cá nhân t/g

+ Chữ ký số (chữ ký điện tử) và một số kỹ thuật mã hóa chữ ký số, giới thiệu các giải thuật SHA [Secure Hash Algorithm] và RSA [Ron Rivest, Adi Shamir và Len Adleman]

Số account quá lớn

+ Công nghệ tìm kiếm và định vị: Semantic Web (Web ngữ nghĩa):

Nền tảng tìm kiếm ngữ nghĩa trên Internet ; VD Google Search

Điện toán đám mây (cloud computing) : lưu trữ, chia sẻ trên mạng all cầu (HDD sẽ bỏ)

GoogleMap API, FaceBook API: khác thác các thông tin d/v vào trong PM

VD: định vị giao hàng, đánh giá tâm lý khách hàng, . .

Kỹ thuật tin nhắn SMS, MMS trong phần mềm: nhắn tin k/h

+ Công nghệ nhận diện và “thị giác máy tính” [Computer vision]

Multimedia = Máy tính có “nhìn” , “nghe” và phân tích đc

Chăm công, soi chiếu, an ninh, điều tiết giao thông, . . .

+ Kho dữ liệu, Khai phá dữ liệu, công nghệ tri thức và ”Tin sinh học” [BioInformatics]

CN tri thức

+ Ngân hàng điện tử (eBanking)

Internet/Phone/Home/SMS Banking

+ “Thương mại điện tử” [eCommerce] và “Thanh toán điện tử” [ePayment]

Mua/Bán trực tuyến (Online) / hành chính công trực tuyến

+ ERP và CRM

Enterprise Resource Planning

Customers Relation Management

+ Dịch vụ SEO = Search Engine Optimizer

+ E-Learning và Distance Learning

+ Báo điện tử [ePaper] và RSS /// báo điện tử = tóm tắt luồng tin / chủ đề

+ Cloud Computing, Phone App (Mobile App ) App Store

### **Câu 3: Các giai đoạn cơ bản trong tiến trình phát triển hệ thống phần mềm? và các bước cụ thể trong các giai đoạn đó? minh họa đơn giản cho mỗi bước?**

HD:

2 giai đoạn :

[1] Xây dựng kiến trúc hệ thống phần mềm

PT KTPM

TK KTPM

[2] Thiết kế hệ thống phần mềm

1. GD

2. DL

3. XL

4. BM

5. Help

6. Test Case

7. Giao tiếp khác = SMS, định vị, CV, Scancode/QR code, eBanking, . . .

8. Cài đặt = Setup

9. Đặc thù : PhoneApp, ĐK tự động

Chú ý:

+ KT => TK

+ Q/H

+ ko bb thực hiện all cv nêu trên đ/v all pm ; thứ tự tùy tr/h cụ thể [cân nhắc cho phí...]

2. Giới thiệu một Công cụ hỗ trợ phân tích, xây dựng kiến trúc và thiết kế phần mềm

. Code Map / [Architech]. Visual Studio.net (Enterprise)

. Component Diagram + Deployment Diagram /UML (Rational Rose)

### **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, thương phải có Hệ thống kiến trúc phần mềm như thế nào? Và với Hệ thống kiến 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**

HD

\* 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

KHI THIẾT KẾ KTRUC CÁC HT PM, CÓ THỂ :

- XD HTPM DẠNG WEBAPP & mobileApp: CÓ 2 HT HĐ SS (LOCAL + INTERNET)

KHI ĐÓ PHẢI THIẾT KẾ CÁC CHỨC NĂNG:

+ ĐỒNG BỘ HÓA :

+ PHÂN BIỆT CÁC TR/H NSD:

**Câu 5: Môi quan hệ giữa Kiến trúc phần mềm và Thiết kế phần mềm? Minh họa thực tế? Có cần phải Thiết kế kiến trúc phần mềm cho tất cả các trường hợp phần mềm hay không, vì sao? || Chỉ Thiết kế Kiến trúc phần mềm đối với những trường hợp phần mềm như thế nào?**

HD

+ chỉ pm qui mô

+ quan hệ: kt & tk

**Câu 6: Các yêu cầu & nguyên tắc cơ bản về TK PM? Minh họa.**

HD

Chú ý: 8 nội dung TK : GD, XL, DL, BM, TG, Giao tiếp, Cài đặt, Test

Nguyên tắc:

+ Lấy người sử dụng làm trung tâm (y/kiến nsd, tham vấn kỹ thuật,...>thống nhất)

+ Phải đảm bảo NSD sử dụng triệt để các chức năng of PM:

có hệ thống trợ giúp tốt (helper):

. Tổng quan.... : (.NET) : HelpProvider

. Hỗ trợ tại mỗi bước đảm bảo hạn chế thao tác sai ...: ToolTip

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

. ....

+ hỗ trợ người dùng có bản quyền, . . .

+ hiệu quả kinh tế và ứ dụng = công + chi phí <= kq đạt (...)

....

Yêu cầu:

+ Cho phép nsd chọn 1 trong nhiều phương án thực hiện 1 thao tác nào đó: VD để in vb trong word: [Menu], Toolbar, Ctrl \_P, ..=> xác định được dạng lựa chọn của nsd thành thạo [Hotkey] & người mới s/d [Menu]

+ Thiết pm phải đảm bảo tập trung ưu tiên phù hợp cho nsd thành thạo

VD: màn hình tờ khai HQ: 2 p/a giải quyết: ghim / checkbox

+ hướng đơn giản cho nsd, tránh gây rối: ko nên quá nhiều màu sắc và ko chọn các m.sắc gây áp lực nsd

+ có sự thay đổi đột phá, tránh gây nhầm chán cho nsd khi sd lâu dài

+ đảm bảo tâm lý tốt cho nsd: vd

....

**Câu 7:**

**a) TK các hình thức thực hiện một thao tác nghiệp vụ trên pm, ưu tiên cho những nsd nào ? vì sao? Minh họa thực tế?**

**b) xét thao tác in văn bản trong pm word, theo a/c nên thiết kế những pp nào để t/h thao tác trên ? pp nào đc ưu tiên hơn? vì sao ?**

**a) Xét pm uis.ptithcm.edu.vn với thao tác: Đăng nhập, theo A/C nên thiết kế những pp nào để nsd lựa chọn và thực hiện? PP nào ưu tiên hơn ? vì sao?**

HD: a)

- Các khuynh hướng thiết kế hình thức thao tác cho người sử dụng

+ Thiết pm phải đảm bảo tập trung ưu tiên phù hợp cho nsd thành thạo

VD: màn hình tờ khai HQ: 2 p/a giải quyết: ghim / checkbox

,

## Câu 8: Các bước tương tác cơ bản của nsd trên ht pm? Minh họa thực tế? Có bắt buộc mọi thao tác có đầy đủ 4 bước (VD)? Bước nào q tr nhất? vì sao (minh họa)?

HD:

Các bước thao tác of nsd: 4 BƯỚC

B1: CHỌN “LỆNH” / THAO TÁC CẦN THỰC HIỆN; VD:

B2: CUNG CẤP TT CẦN THIẾT ĐỂ PM T/H “LỆNH..; VD:

B3: RA LỆNH CHO PM THỰC HIỆN “LỆNH:/TT CẦN THIẾT SAU KHI CC ĐẦY ĐỦ TT

B4: NSD TIẾP NHẬN KẾT QUẢ T/H “LỆNH:/TT NÊU TRÊN: 2 K/NẮNG:

ĐẠT Y/C

KO Y/C = LỖI : PM HỖ TRỢ NSD K/PHỤC LỖI

Chú ý:

+ các bước nêu trên là 1 tr/h “kịch bản” khai thác sd pm của nsd trong t/h đúng nhất

Tuy nhiên trên thực tế nsd có thể t/h theo “kịch bản” khác; vd: “thêm”->”xóa”,...

Kịch bản gây lỗi-> test

+ ko nhất thiết all các tt đều có đầy đủ 4 b nêu trên, ko nhất thiết theo trình tự:

Vd: XÓA DA: 2-4>: chọn da cần xóa[b2]->xóa[b3]->xem kq[b4]

SỬA DA: chọn da[b2]->sửa[b1]->nhập tt mới[b2]->”đồng ý”[b3]->b4

+ B3: Quan trọng 1->lập trình, “lệnh” đc chính thức t/h

## Câu 9: Khi tk HT Pm WebApp cần quan tâm vấn đề gì? Minh họa thực tế? khi x.đ múi giờ thao tác của nsd ta chọn múi giờ phía Client / Server ? vì sao ?

- Cường độ truy xuất pm là giống nhau trong các giờ trong ngày (đêm = ngày) / <>WinApp

- Xác định múi giờ cập nhật / xử lý trên ht pm sao cho phù hợp trong các giao dịch của ng sd (tùy PM WebApp):

vd: thông tin thời điểm đặt hàng trên webapp bán hàng trực tuyến: mua (client)

hay bán (server)? + thiết kế chức năng chọn múi giờ

Chẳng hạn: cty bán hàng online vn lúc 8g s 08/06/22[Server] có 1 kh usa đặt hàng mua tại thời điểm đó (usa) là 23g 09/06/22[Client] => trên đơn of kh nay ghi ngày đặt hàng là ngày nào (08/6 hay 09/06)? Vì sao? [ko có trong WinApp]

- Thiết kế GD Pm dạng đa ngôn ngữ (ưu tiên ngôn ngữ tại những nơi nsd nhiều nhất) + khi tke các chức năng gd, dl , xl => phong tục/ tập quan / tôn giáo / tín ngưỡng / pháp luật of các cộng đồng nsd / quốc gia liên quan

- sử dụng kỹ thuật đám mây (cloud computing) & đồng bộ thông tin / dữ liệu

## Câu 10: các nguyên tắc & y/c cơ bản trong TK GD PM? Minh họa thực tế?

\* Nguyên tắc cơ bản: (ngoài NT TK PM nói chung)

[1]- NSD .,:

Vd: button, textbox, ...: controls (NSD sẽ dùng những đt này để điều khiển PM

Vì đc xem xét trên góc độ nsd = các đt nêu trên giúp nsd điều khiển ht pm

[2]- XD các đ t nsd gd đó :....(nhiều đt cùng s/d gd)

. mỗi nsd quyền lv riêng / form sẽ tk, cần xd nsd trước khi tk

. với 1 form gd có thể có nhiều nsd l/quan

Vd: gd nhập điểm thi sv: gv, gv khoa, cb khảo thí

Sv: không được phép truy xuất

=> nếu tổ chức form dạng web / internet : BB bảo mật

[3]- Thiết kế gd trong tổng thể các thiết kế khác; như: bảo mật, trợ giúp, giao tiếp, cài đặt, kiểm thử,...; không tk gd một cách độc lập [KIẾN TRÚC PM ĐÃ CÓ]

VD: tk form nhập điểm liên quan bảo mật theo nsd: GV được quyền gì, sv không được truy xuất, ...

[4]. XD loại Form GD :

. Vì mỗi loại form gd có những đặc trưng riêng => á hưởng đến tk

. Mỗi CN PM có thể có nhiều loại Form (vai trò tương đương hoặc KO tương đương):

VD: GV Nhập điểm: Web, Mobil : tương đương

Thư ký Khoa kiểm do: Win, Web: Win chính, Web: phụ = ko tương đương

VD:

+ webform:

. KHÔNG thoát: sign in + sign out

. paging

. không gian tk tương đối rộng hơn so với mobile; tuy nhiên phụ thuộc vào loại trình duyệt và độ phân giải => khắc phục bằng Table và thường có 2 phiên bản: PC và Pocket: vd [www.facebook.com](http://www.facebook.com) & [m.facebook.com](http://m.facebook.com)

. cơ động, nhưng chỉ khi có PC

. Không có messagebox riêng (thường nhờ web browser): thay thế bởi Label, . . .

....

+ winform:

. thoát (this.close(), Application.Exit(), Me.Close()): dùng chọn

. scrollbar

. không gian thiết kế gd ổn định, rộng rãi hơn so với mobile

. chỉ phù hợp nsd văn phòng / nhà : cố định

. Có MessageBox

...

+ mobileform:

. thoát : dùng cảm biến

. scrollbar

. thiết kế gd hạn chế => phải cân nhắc tránh lạm dụng

. NSD cơ động, vì dùng SMART; phù hợp nsd di chuyển thường xuyên; vd nv shipper

. MessageBox trên nền form

...

[5] TK các t/p of Form GD cần đóng vai nsd: chỉ biết NV, ko chuyên công nghệ

Vd: xét form gd quản lý kho hàng

Button sau là ko hợp lý:

“Thêm kho” “Xóa kho” “Sửa kho”: do kỹ thuật csdl

Nsd ko hiểu, ko chuyên csdl

[6]- ưu tiên nsd thành thạo

Vd: xét form gd quản lý khai HQ

[7]- cần phân tích các nghiệp vụ trên giao diện trước khi thiết kế gd: => x/d các y/c cụ thể of all ht gd & từng gd => test pm sau này (liên quan Tester)

[8]- Tương đối hóa các giao diện: ko nên quá “cầu toàn” các xl trên giao diện,

vd: để an toàn, có quá nhiều các thao tác bị cấm,

Ko báo những lỗi ko quá cần thiết....

...

[9]- Cho phép NSD có nhiều phương án thay đổi giao diện

(Option, Customize,...)

\* Các yêu cầu về GD PM:

.Tùy thuộc vào từng loại gd, dt sd, loại nv làm việc trên gd ...;  
y/c chung cơ bản; các gd đ/ư

*Vd: nv vp = ..... = nhiều, phức*

*Bếp, shipper .... = chọn*

- dễ s/d ... (ko học nhiều)

- gắn liền qui trình nv : form, report phù hợp biểu mẫu, đảm bảo tính logic (có thể tham vấn nsd)

- đơn giản => ko gây khó chịu nsd; VD ko quá nhiều màu sắc, tránh màu sắc nóng (đỏ, cam, vàng,...) => chú ý tâm lý nsd;  
vd ko nên cho sửa msnhom đa / form ql đa -> nsd cảm giác mất dl

- trực quan các thao tác of nsd: WYSIWYG

VD Giúp nsd xd vị trí họ làm việc : đã làm, sẽ làm tiếp ?

VD tham khảo Form khai báo HQ

- Cho phép NSD có nhiều phương án thao tác trên giao diện : VD tổ hợp phím (thành thạo), thực đơn, toolbar, Contextmenu, VD: tìm DA qua nhiều thông tin khác nhau (mã da, tên da, cdt, ...)

CHÚ Ý: TK TK P/A LỰA CHỌN NSD T/H TT / PM

- 3 P/A : MENU, TOOLBAR, HOTKEY

- KO ALL,

o CĂN CỨ:

▪ CẤP ĐỘ NSD:

▪ TẦN SUẤT S/D TT:

- ko dư thừa không gian giao diện ; VD Win8

- bố trí các thành phần trên gd sao cho: những thao tác quan trọng = thường dùng => đặt tại các vị trí dễ nhìn, quan sát (gây chú ý nsd); các thao tác có rủi ro cao, nguy hiểm dữ liệu, phần mềm, đòi hỏi chuyên môn cao,... => bố trí những vị trí thích hợp => hạn chế sai sót, rủi ro,...

VD form QL DA: xóa tránh thao tác nhầm, Thêm+Sửa: đổi nhãn (thay vì dùng button mới); Enable, Visible thiết kế các Button tránh thao tác nhầm

- Nhất quán về dạng các giao diện & thành phần trên giao diện: VD thông báo lỗi (một dạng form với màu đỏ,...)

.....

Chú ý: ngoài các y/c cơ bản, có thể có các y/c chi tiết khác; tùy

+ Loại form: vd: winform, webform, mobileform,...

+ Người sd: vd: thành thạo, mới sử dụng

+ Nghiệp vụ liên quan: VD:

thư ký -> phím, nhân viên nhà hàng thì -> chuột

TÙY GD, Y/C KHÁC CỤ THỂ HƠN

VD: WebForm => nhiều t/b + webbrowser khác nhau : ePaper...

MobileForm => (không gian gd hẹp) => tiết kiệm thành phần gd: ko quá nhiều button, label,...

...

## **Câu 11: VÌ SAO KHI TK CÁC HTPM QUAN TÂM ĐẾN HTGD?**

**HỎI KHÁC (tương đương)- KHI TK HT PM, LẤY HTGD LÀM TRUNG TÂM ? VÌ SAO ? PHÁT BIỂU NÊU TRÊN CÓ MÂU THUẪN GÌ VỚI P/B : “KHI XD PM PHẢI LẤY NG SD LÀM TT”, GTHỊCH?**

**HD: . P/V NSD, L.V PM QUA GD => TK KHÁC (DL, XL,...) X/P TỪ GD**

**. TR/B, XD Ý TƯỞNG, K/S, PT HT PM : DỰA TRÊN CS CÁC PHÁT THẢO GD**

## Câu 12: Để đánh giá chất lượng of ht màn hình gd, thường phải đảm bảo những nguyên tắc gì? và căn cứ những nhóm tiêu chí nào ? Minh họa thực tế?

NT:

- nsd làm trung tâm

- xem xét / 2 phương diện : sản xuất [kỹ thuật] + sử dụng [hiệu quả]

- Không có bộ tiêu chí nào là chung cho all GD

VD cùng 1 PM nhưng tiêu chí đánh giá có thể khác nhau tùy vào: dạng/loại pm, mục đích sd, đối tượng sd, người đầu tư

- Xác định trước khi xây dựng pm: xác định Giới hạn phạm vi & xđ tiêu chí đánh.

. bộ tiêu chí đ/g cl sp pm: trước khi thực hiện việc sx pm (liên quan : case kiểm thử );

- Thay đổi theo time do yếu tố công nghệ kỹ thuật phát triển

... .

\* đ/g (mh gd) nhóm tiêu chí gợi ý như sau:

. ngoài các tiêu chí đương nhiên (vd: chạy đc, đúng, đủ,...);

Đối với đơn vị sản xuất PM: kỹ thuật

đ/v sử dụng PM: hiệu quả

Đầu tư t/c

Trên những nhóm

+ Tính đúng đắn: so với phân tích, thiết kế ht pm & các yêu cầu cụ thể đã đặt ra

+ Tính tiện dụng: . sử dụng dễ nhất, nsd ít tốn thời gian để học cách dùng. Nsd có khả năng khai thác sử dụng hết tất cả các tính năng của pm có trên gd

+ Tính hiệu quả:

. ít tốn: công sức, chi phí, t.gian khi thiết kế gd

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

. ng sd ít thao tác, ko lặp những thao tác giống nhau, hạn chế sai sót khi thao tác trên gd

. gd được tk sao cho => việc sản xuất pm ít tốn nhất (code đơn giản, ...)

*TK giao diện ==> ảnh hưởng ==> lập trình*

... .

+ Các tiêu chí khác, tùy dạng pm

. Bảo mật (Security): tranh truy xuất thông tin / pm bất hợp pháp [Hacker / Hacking]: VD

. An toàn (Safety) thao tác: tránh hư hỏng / mất thông tin / dữ liệu [Virus, Backup / Restore,..]

, ..

## Câu 13: Giới thiệu một số dạng màn hình giao diện thông dụng [tên gọi, ý nghĩa s.d, nội dung] ? Một màn hình gd có thể thiết kế theo nhiều trong số các dạng nêu trên đc ko? vì sao? Minh họa thực tế?

HD:

\* Hướng tới fr GD mở, đặt biệt kết hợp AI (ML) => form động = tự thay đổi theo thực tiễn sd

+ GD mở = thay đổi được theo thực tiễn, thậm chí là tùy ý thêm bớt [NSD]

GD mở = tùy biến theo thực tiễn;



\* Sau là 1 số form truyền thống (WinForm, ...)

Loại GD	Ý nghĩa Sđ	Nội dung chính
MH chính (Winform=MainForm Webform=HomePage)	Chọn C/v: Form	D/s các C/v
Nhập DL lưu trữ	Lưu trữ t/t mới: Form	Form nhập DL: D/s các thông tin phải nhập; Control lưu trữ
Nhập DL xử lý	Nhập thông tin cho phép thực hiện 1 thao tác khác (có thể chuyển điều khiển sang form khác):Form	2 phần: - D/s thông tin nhập= input - Đ/k chuyển hoặc XL
MH Kết xuất K/quả (Report)	Trình bày K/q cho NSD (Reports)	D/s các thông tin K/q cần kết xuất
MH thông báo trạng thái (MessageBox)	Thông báo Trạng thái HT, cảnh báo/nhắc NSD thực hiện c/v	D/s nội dung / hình tượng cần thông tin đến NSD
MH Tra cứu = tìm kiếm (Search= Find = Locate)	Nhập thông tin cơ bản, tra cứu các thông tin khác đã lưu trữ <u>Chú ý:</u> Cá biệt nhập DLXL	2 Phần: - Nhập các t/t cơ bản - T/t Kết quả tra cứu (ds)

Chú ý:

- tương đối : 1 MH GD thuộc nhiều loại

- Khuyh hướng hiện nay : thiết kế các MH GD theo cơ chế mở = tt / xl trên mhgd linh hoạt, tùy biến theo cách của nsd và theo tình huống

- tùy loại mhgd => biến thể of các dạng mhgd nêu trên:

vd : webform form sơ đồ websites = sites map path / link tới các trang tw theo nút trên cây

**Câu 14:** so sánh (phân biệt) : *[những khác / 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.

HD

+ webform:

. KHÔNG thoát: sign in + sign out

. paging

. không gian tk tương đối rộng hơn so với mobile; tuy nhiên phụ thuộc vào loại trình duyệt và độ phân giải => khắc phục bằng Table và thường có 2 phiên bản: PC và Pocket: vd [www.facebook.com](http://www.facebook.com) & [m.facebook.com](http://m.facebook.com)

. cơ động, nhưng chỉ khi có PC

. Không có messagebox riêng (thường nhờ web browser): thay thế bởi Label, ...

....

+ winform:

. thoát (this.close(), Application.Exit(), Me.Close()): dùng chọn

. scrollbar

. không gian thiết kế gọn ổn định, rộng rãi hơn so với mobile

. chỉ phù hợp nsd văn phòng / nhà : cố định

. Có messagebox

...

+ mobileform:

. thoát : dùng cảm biến

. scrollbar

. thiết kế gọn hạn chế => phải cân nhắc tránh lạm dụng

. NSD cơ động, vì dùng SMART; phù hợp nsd di chuyển thường xuyên; vd nv shipper

. MessageBox trên nền form

=====...

**CÂU HỎI NÀY, LIÊN QUAN ĐẾN BÀI TẬP**

BT: CHUYỂN FORM : VD: CHO WINFORM =&gt; WEBFORM ? VÀ NGƯỢC LẠI ( &amp; cả MobileForm)

VD: BÀI TẬP 5 (BẮT BUỘC 2) :

+ Đề bài Winform = Hãy thiết kế lại Form đã cho theo dạng WebForm OR MobileForm?

OR

+ Chụp Form (trích từ App) dạng Mobile (Web) =&gt; y/c design lại theo 2 dạng còn lại

**Khi chuyển thể THIẾT KẾ một Form từ dạng này sang dạng khác, sẽ liên quan và ảnh hưởng đến ALL thiết kế khác (không chỉ UI)**

[8 ND TK: GD, XL, DL, GT, CĐ, BM, TG, TEST]

. VD: gốc WinForm => WebForm -> phát sinh vấn đề Bảo mật cần phải thiết kế bổ sung; vì: nếu Winform thì thường là s/d trên PC nên không quá quan tâm đến Bảo mật; nhưng với WebForm thường là sd trên Internet (toàn t/g) sẽ phát sinh bảo mật; ngoài ra còn liên quan đến các tk khác, vd: trợ giúp, giao tiếp ngoài với Camera Qr code, ...; như: DL phải Ditrubution; Test thiết kế lại như Load Test; cài đặt Setup-> Up Host

**CHÚ Ý KHI TK MHGD**

+ winapp: tận dụng ko gian mh =&gt; vd: win8

+ mobileapp: không gian hạn chế; nsd cơ động (VD: NV Giao hàng)

+ web: quan tâm đến việc đa dạng giao diện = phù hợp với nhiều dạng trình duyệt khác nhau (Safari, Opera, IE, Chrome, Firefox, ....)

.xd dt s/d = web browser :

PC (www.facebook.com) || MOBILE (m.facebook.com)

. người dùng all ; ko biết; vh pt tq + ngôn ngữ, thuật ngữ =&gt; MultiLaguages

Chú ý: dùng SMART phone truy xuất &amp; sử dụng web (ePaper) =&gt; pm WebApp

PhoneApp = pm tải từ App Store &amp; cài đặt + s/d trên SMART Phone

- Gợi ý:

VD MH chính = WebForm: HOMEPAGE // Winform: MainForm // PhoneApp: MainForm=StartForm

VD HomePage / WebForm:

+ truyền thống =&gt; Dropdown Menu / Ribbon Menu

+ tmđt = DataList

+ form : khác

**Câu 15: Thiết gd ảnh hưởng ntn đến tâm lý nsd ? Minh họa thực tế?****[Hỏi tương đương] Vì sao khi tk dg pm cần quan tâm tâm lý nsd? Minh họa một số trường hợp cụ thể ?**

Gợi ý

\* tk ht gd phụ thuộc trực tiếp vào yếu tố tâm lý nsd (đặc biệt là tk trợ giúp); vì vậy khi tk gd (+trợ giúp), cần q/tâm những y/t tâm lý sau đây of nsd:

1. Nsd ko thành thạo về cnpm &amp; thậm chí thao tác máy tính ko tốt

=&gt; vì vậy, tk gd pm: cần đóng vai nsd ko thành thạo cnpm

VD: Form quản lý Kho hàng:

[Giao diện](#)

Mskho tên kho

01 Q9

02 Q1 &lt;=Hightlight

Thêm Sửa Xóa = Trả kho / chuyển mục đích s/d kho, ...

[Dữ liệu](#)

KHO(mskho, ten kho, ...)

01 Q9

02 Q1 &lt;=XÓA

TONKHO (mskho, msmh, sl)

01 01 3

01 02 5

01&lt;- 02 01 7: sửa mskho=chuyển all MH từ kho 02 -&gt;01

01&lt;- 02 02 4 : sửa mskho=chuyển all MH từ kho 02 -&gt;01

Kỹ thuật CSDL: xử lý RBT/ DL dạng PTH: PriKey-ForK/th

VD2:

MH(msmh, tenhang, ...

01 Áo  
02 Quần

Form “Giỏ hàng” / PM bán hàng Online

Msmh tên hàng dvt gia sl tiền

01 Áo cái 5 4 20: Xóa Sửa  
02 Quần cái 7 3 21: Xóa Sửa

Xóa = Trả hàng = Lấy hàng ra khỏi “giỏ”

Sửa = Đổi hàng[msmh, sl] / Thêm/Bớt hàng trong “giỏ”[sl]

2) Khi nsd gặp phải 1 sai sót / lỗi do thao tác => ngta liên tưởng đến toàn bộ htpm: đánh giá chất lượng pm kém => ngại dùng pm (từ chối s/d).

VD: xét bài tập đánh giá gd qđ dự án: nếu cho nsd sửa mã số nhóm dự án ngay trên Form quản lý dự án => làm cho tên dự án biến mất khỏi ds dự án =tâm lý nsd=> làm cho họ lo lắng: có sự mất mát dữ liệu, trong khi thực tế dữ liệu ko mất, mà chỉ chuyển sang nhóm khác, nên ko còn trong ds nhóm này nữa: xét về kỹ thuật tk gd ko sai, xét tâm nsd ko đạt y/c => nsd ko tin tưởng p/m => ngại s/d pm ..

3) NSD Ngại đọc text thay vào đó họ thường quan sát hình

VD: chọn biểu tượng Cờ Việt/Anh để thay đổi ngôn ngữ trên giao diện, thay vì chữ

...

4) Có thể thay đổi theo thời gian, tránh “nhảm chán” cho NSD, sd lâu dài; tuy nhiên ko làm “bất ngờ” (ko làm NSD e ngại) = ko thay đổi đột ngột.

5) Hình thành sự tin tưởng của NSD vào đ/v sp pm (tương lai), trước khi thiết thông quan các thủ tục về tâm lý; chẳng hạn = giới thiệu trước về tính hiệu quả, tiện dụng, ....=> hình thành niềm tin : => dễ NSD dễ thông cảm & chi sẽ & phối hợp với bên sx pm khi sp pm có phát sinh sự cố kỹ thuật trong tương lai

...

## Câu 16: Trình tự các bước tk 1 Form GD / HTPM ? Minh họa thực tế?

[TRÌNH TỰ TK 1 FORM / HT PM UD: 9 BƯỚC](#)

**B1:** Xác định vị trí các form trong tổng thể KTPM

**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 nsd // dựa vào kiến trúc pm đã thiết kế;

- Loại nào ưu tiên hơn? Cần bảo mật không? Có giới hạn quyền không? Tách (dùng form phụ) / ghép form ....

- Vd: grid-wiew

**B3:** Xác định mục đích chính của Form : làm gì ?

**B4:** 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: KH

**B5:** TKE FORM : GD -> XL [-> DL]....

B5.1: Thiết kế sơ đồ tĩnh của HT Forms & Xác định vị trí trong Sơ đồ tĩnh: mô tả ht tĩnh của gd: thường sd mô hình tree (đồ thị)

+ Mô tả tĩnh = logic khi thiết kế (chtr ko chạy), chưa quan tâm vấn đề chuyển điều khiển giữa các form

+ Chỉ có tên mh gd (và thể có chỉ số nhận diện: 1. 1.1. 1.2. )

VD: WebApp: SiteMapPath

B5.2. Phát thảo: hình thức, vị trí các thành phần trên form (kể cả các nút lệnh điều khiển), màu sắc,...)

+ Lần lượt thiết từng Form gd trong mối quan hệ giữa các form gd theo sơ đồ cây nêu trên

B5.3. Kịch bản khai thác : DFD tĩnh gọn || Activity Diagram/UML: Thiết kế các sơ đồ kịch bản khai thác sử dụng các form gd

- Thường dùng sd DFD tĩnh gọn (ko có kho dl)

- Mô tả tr/h NSD thao tác và làm việc đúng các nghiệp vụ / form

VD: khi nsd chọn nhóm “Tất cả” [mã \*] và Thêm MH;

->DFD không thể hiện được lỗi này của nsd

Kết hợp sd chuyển trạng thái Petri

+ Trong form gd (thay đổi trạng thái form): Thêm->NSD nhập ..

+ Chuyển sang form gd khác: VD fr:QLHH-thoát->MH Chính

VD: khi nsd chọn nhóm “Tất cả” [mã \*] và Thêm MH;

->sử dụng Petri để thể hiện được lỗi này của nsd

Chủ yếu là nút lệnh điều khiển:

- Kịch bản xử lý phát sinh lỗi: Ptri (tìm những trường hợp cơ bản; KHÔNG thể tìm hết được)

Mô tả tr/h NSD thao tác và làm việc sai (cá biệt: vd nhập thiếu dl ) các nghiệp vụ / form

Kịch bản xử lý phát sinh lỗi: Ptri (tìm những trường hợp cơ bản; KHÔNG thể tìm hết được)

**B6: BẢO MẬT :** QUYỀN TRX OF TỪNG Đ/T => .ENABLE / .VISIBLEvd:

**B7: TK trợ giúp** / Form và liên Form

**B8: T/K CƠ CHẾ + T/P ĐÓNG GÓI, CÀI ĐẶT, CHUYÊN GIAO** : TỰ ĐỘNG [ FONT, kết nối db,...], BẢN QUYỀN + HỖ TRỢ NSD có bq, ...

**B9: TK case Kiểm thử [Unit Test]** (nếu cần) => beta

+ Ngoài ra, có thể phải Dự kiến về khả năng mở rộng PM trong tương lai  
(do nhu cầu NSD OR sự thay đổi or phát triển các nền tảng KH CN)

**Chú ý:** KO phải lúc nào cũng thực hiện đầy đủ 9 bước

### Câu 17: Các cấp độ TK HT Form GD of PM? Trình tự thực hiện trong thực tế ntn? Minh họa thực tế?

**CẤP ĐỘ TK FORM / HTPM: 3 CẤP ĐỘ**

C1: TÙNG FORM (9 BƯỚC: NÊU TRÊN)

C2: MODULE OF NHÓM C/NĂNG (PROJECT) = Components

C3: ALL HT PM (SOLUTION)

Thực tế thực hiện: C3->C2->C1

### Câu 18: Giới thiệu một số hình thức & dạng kết cấu mục chọn thực đơn thông dụng hiện nay trong các htpm? Các Nguyên tắc TK thực đơn PM? Minh họa thực tế?

\* Nhiều hình thức, hiện nay : 2 hình thức thông dụng

+ Dropdown (Kéo xuống): MenuItem (mục chọn)

VD: MS. Visual Studio . NET

Vnexpress.net

+ Ribbon (Ruy băng)

VD: MS. Word

<https://tuoitre.vn>

...

\* Nhiều dạng kết cấu, thông dụng: Kết cấu OO hoặc FO

OO = đối tượng = Object-Oriented = ....

FO = chức năng = Function-O... =

TK kết cấu các mục chọn Menu

OO: B1: x/d ds các đt ql of ht pm (thực tế): VD QLSV: SV, MH, Lop, ... (noun)

|

B2: Nhận diện các đối tượng trung tâm ql of PM: ...

|

B3: X/d nsd của Menu: nv p.đt, sv, gv khoa, gv, ... & quyền truy xuất

|

B4: TK mục chính of menu = ĐT trung tâm (B2) & quyền s/d (B3)

|

B5: TK các mục chọn of từng thực đơn (MenuItem, Ribbon)

FO: B1: x/d ds các CHỨC năng of ht pm: VD QLSV: Tuyên sinh, đào tạo, thống kê báo cáo, ... (verb)

|

B2: Nhận diện các chức năng trung tâm ql of PM: ...

|

B3: X/d nsd của Menu: nv p.đt, sv, gv khoa, gv, ... & quyền truy xuất

|

B4: TK mục chính of menu = CN trung tâm (B2) & quyền s/d (B3)

|

B5: TK các mục chọn of từng thực đơn (MenuItem, Ribbon)

Chú ý:

+ OO + FO

+ Mục chọn menu: tĩnh (cố định) / động (thay đổi bởi nsd); tĩnh (tương đối = nsd tùy biến / customize) || SMART Menu dùng AI & ML

+ Menu thông thường / màn hình chính (WinForm/Mobile: MainForm| StartForm, WebForm: Master/HomePage)

\* Nguyên tắc thiết kế menu of HT PM

1- Các mục chọn of menu: có đầy đủ all c/n (command) pm c/c cho nsd

2- Gọi nhớ và phù hợp thực tế s/d pm: phân chia nhóm menu và đặt tên các mục chọn menu giúp nsd 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 menu càng đơn giản càng tốt; hạn chế trùng lặp, trừ 1 số trường hợp do chức năng s/d có liên quan đến nhiều nhóm menu

4- TK kết cấu menu ưu tiên theo nsd chưa thành thạo (Beginner)

5- Linh hoạt = có nhiều tùy biến và thay đổi theo từng đối tượng nsd

6- Ko chiếm nhiều không gian nhưng phải tập trung đc chú ý of nsd

-> tùy từng trường hợp đặc thù pm, có ng tắc khác

'===TK MainMenu của PM QL DA VT (BT 5= BB2); Menu ?===

VD: [Hệ thống] [Trình Bày] [Dự Án] [Nhân lực]...

MainMenu của MS. Word là OO or FO?

=====

đánh giá ưu \_ nhược:

+ Dropdown (Kéo xuống): MenuItem (mục chọn)

+ Ribbon (Ruy băng)

=====

THIẾT KẾ THỰC ĐƠN CHÍNH CHO PHẦN MỀM QUẢN LÝ ĐÀO TẠO TẠI CÁC CƠ SỞ ĐẠI HỌC VN?

YÊU CẦU:

[1] Xác định dạng (loại) thực đơn sẽ dùng

[2] Thiết kế các Thực đơn chính & các mục chọn cơ bản của từng thực đơn chính

[3] Nhận xét & đánh giá : hệ thống thực đơn đã thiết kế

## Câu 19: Giới thiệu một số dạng đa dạng hóa hình thức trình bày các form? Minh họa thực tế?

HD

+ Cơ chế form thay đổi tùy vào tình huống và đk sd của người dùng

+ Đặc biệt là Webform, vì số toàn thế giới với ALL đối tượng NSD , ko cần biết NSD là ai

=> khi thiết kế các form phải thực hiện đa dạng theo ngôn ngữ, thói quen, phong tục, tập quán, mục tiêu, luật pháp,...

+ Khuynh hướng tương lai là form tự tùy biến & đa dạng dựa vào AI = Form có kn tự biết nhu cầu NSD và điều chỉnh theo (vẫn cho NSD chủ động chọn)

+ Công cụ hỗ trợ thiết kế & lập trình Form: hiện nay chưa đáp ứng, trong thời gian tới sẽ có hỗ trợ.

VD:

- Có nhiều pp: XML file đa dạng nhãn ngôn ngữ

VD: để chọn V và E s/d 2 files XML

V.xml (nhãn tiếng Việt)

E.xml (nhãn tiếng Anh)

s/d các Tag giống nhau (save as = Sửa content)

⇒ Code : using xml theo các lựa chọn of nsd (cờ)

Bt1.Text = <nhân 1>

**Câu 20: Tối ưu hóa form gd là gì [mục đích , tiêu chí ]? Tách/ghép các form gd phải đảm bảo những nguyên tắc gì ? Minh họa thực tế?**

**Câu 21: HÃY CHO 1 VD VỀ VIỆC TÁCH/GHÉP FORM/PAGE CÓ ẢNH HƯỞNG ĐẾN MÃ LỆNH (CODE)**

**HD**

- **Mđ = đảm bảo tính hiệu quả của form gd**

Tách/ghép form: ảnh hưởng các tk khác : code, help, security, giao tiếp khác, db, test, setup,... => đơn giản

Vd: xét thêm d/a / bt5 / tr5:

p/a 1: thêm dự án : pp đổi nhãn: “thêm...” => “Luu...”

[1] codes phức tạp : if(nhận diện: Thêm...)

{ ...chuẩn bị thêm...}

else(nhận diện:Luu...)

{ ....luu d/a mới vào DB...}

[2] help, security: tk lại -> phức tạp

Tránh tr/h nsd thao tác vô lý, vd: “thêm...”->”xóa...”/”sửa...”

Giải pháp: .enable = false: vị trí ko đc chọn

...

p/a 2: thêm form phụ (tách form...) thêm dự án

tăng sl form, các tk khác đơn giản hơn

[1] codes đơn giản: form chính: Thêm...

...chuẩn bị thêm...

Form phụ: ....Luu...

....luu d/a mới vào DB...

[2] help, security:

Nsd tập trung, gần như ko nhầm ..

Nsd truy xuất ko hợp lệ

Test hộp trắng dễ hơn

...

**Tiêu chí = Xem lại câu 12**

**Tiêu chí khác nhau : NSD, TỐC ĐỘ, BỘ NHỚ, ...**

**Chu ý: NSD-THỰC TẾ THÔNG QUA K/S NSD FREE [TEST, BETA]**

- **Tối ưu hóa gd** = có nhiều pp = tách/ghép form là 1 trong những pp, ngoài ra còn có: tái cấu trúc các t/p trên gd, ...

- **Tách gd có nhiều dạng** = bổ sung các form phụ, liên kết hỗ trợ form; or sd xml vd: ....

**Tách/ghép Form** =ảnh hưởng => Codes || KO ảnh hưởng DB

**VD1: Form QLDA : Thêm = form phụ**

**Câu 22: TRÌNH TỰ THỰC HIỆN việc thiết kế hệ thống form giao diện theo theo HT DB đã có?****Minh họa****HƯỚNG DẪN TRÌNH TỰ THỰC HIỆN BÀI LÀM (tham khảo)****B1:** Phân tích & xác định tính hợp lý của DB: chủ yếu vấn đề chuẩn hóa (3NF)

Nếu chưa =&gt; chuẩn hóa

[Các “quan hệ” thuộc tính không khóa PTH đầy đủ &amp; ko bắt cầu vào tập khóa chính, đơn trị]

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

VD: + do msnv trong DDH =&gt; Table NV

+ Điểm trong KH =&gt; cơ chế cập nhật điểm như thế nào

+ Do có giao hàng =&gt; Khu vực &amp; chuyên môn giao hàng

+ Hàng phải lưu KHO và TONKHO

....

Chú ý: Phải xác định phạm vi DB

VD: NV -&gt; Tiền lương -&gt; PM Kế toán

Tài sản cs vật chất -&gt; PM QL TS

NV -&gt; PM QLNS

....

=&gt; Thiết kế thành phần Giao tiếp khác: thiết bị ngoại vi, Các HT PM khác: PM kế toán, QL NS, QL TS, PM QL Marketing, ..

**B3:** XÁC ĐỊNH CÁC ĐỐI TƯỢNG SỬ DỤNG LIÊN QUANKhách hàng ; Nhân viên Quản lý HH, Trưởng bộ phận giao hàng, NV giao hàng, Thủ trưởng đơn vị, ... => Quyền liên quan từng đối tượng sd // **XĐ PHẦN MỀM PHẢI TRONG TỔNG THỂ CỦA 1 KIẾN TRÚC PM=>** do đó, tùy tr/hợp phải phát thảo Kiến trúc of HT PM liên quan: định phân tích PM đang Thiết kế là ở đâu trong tổng thể Kiến trúc PM đó [xđ PM hiện tại giao tiếp với các HT ngoài nào: PM KT, PM 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 DL 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:** VỀ PHÁT 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 phát 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: KH
2. Mục đích chính của Form : để làm gì ?  
VD: Chọn hàng, đặt hàng, mua hàng
3. Loại (các loại) cần dùng : WinForm, WebForm, PhoneForm?  
Loại nào ưu tiên hơn? Cần bảo mật không? Có giới hạn quyền không? Tách (dùng form phụ) / ghép form ....
4. Các Table đọc DL lên? Ghi DL xuống : ... ?  
Input, Trên form, Out ?
5. Các xử lý chính trong Form, chuyển Form? “Kịch bản khai thác”: đúng, phát sinh lỗi,...
6. Màu sắc, hình dạng, xử lý đặc thù ...

**Câu 23: Một số yêu cầu và nguyên tắc cơ bản về TK TP CSDL của hệ thống phần mềm**Một số yêu cầu và nguyên tắc cơ bản về CSDL của hệ thống phần mềm

NT: Phù hợp = [Ko dư, dự kiến mở rộng, Ko thiếu, Ko lạc hậu,...]

Chú ý: “Không lạc hậu” = hầu hết DB của App hiện nay đều triển khai trên Internet (cloud) =&gt; Phân tán, DWH[Dim, Fact], DMN[AI], BigData, ...

YC

+ 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 &amp; vai trò XML

+ Phân tán &amp; 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: Môi quan hệ giữa giao diện và dữ liệu? Các bước thiết kế DataBase theo giao diện cho trước [Câu 22]**

DB =&gt; GD

GD =&gt; DB

+ DB luôn mở rộng =&gt; GD phải mở

+ DB phù hợp nhiều dạng GD

GD =&gt; BD

+ Xét tính hợp lý của GD trước, =&gt; cải tiến

+ Xét tính mở rộng của GD

TK DB

+ Thiết kế GD &lt;== quyết định, ảnh hưởng ==&gt; Thiết kế DL

. Nếu TK DB chặt chẽ, rõ ràng,...=&gt; thiết kế GD sẽ rõ [đơn giản], chuẩn và hợp lý

. Nếu TK GD đơn giản, rõ, hiệu quả,...=&gt; TK DB rõ và chuẩn,...

+ Minh họa:.....

## **Câu 25: Các nội dung Thiết kế hệ thống CSDL cho 1 HTPM :**

1. Thiết kế DB nghiệp vụ = liên quan trực tiếp App

2. Thiết kế hệ thống CSDL bảo mật

Có 2 phương án:

[1] Thiết kế CSDL bảo mật riêng (tách biệt với CSDL nghiệp vụ)

[2] Thiết kế CSDL bảo mật chung với CSDL nghiệp vụ

+ *RBTV = 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

+null

. nsd bỏ qua nhiều tt khi nhập db (fields: allow null) => records quá nhiều null => chất lượng db kém, ko có ý nghĩa sd

Cách xử lý

+ xử lý: -> lọc bỏ (rr)

-> tt nhân tạo (AI) => phỏng đoán các gtri null

-> ngoại lệ ; VD: khách hàng vắng lai

6. Thiết kế xử lý “khóa” trong CSDL

+ mở rộng; vd

+ gợi nhớ = quản lý = xử lý PTH ; VD mã ID công dân

+ Gọn (đơn giản) => dễ nhớ

.....

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

\*) LT MetaData:

8. Thiết kế xl kho dữ liệu cho phần mềm

\* Khi thiết kế PM có sử dụng DB thì theo t/g, sẽ phát sinh lượng db lưu lớn (có thể lên triệu triệu records): dẫn đến vấn đề trên PM

- ko tr x dc OR Chậm => vấn tin DWH, Integration

- Records quá cũ ko có trị sd => phải XL: backup or delete (nếu dễ: nặng, khai phá -> kq sai lệch so thực tiễn)

- ...

=> phải chú ý thiết kế XL DWH

Khai phá dl

XL: khai khoáng, AI

.....

## **Câu 26: Nguyên tắc & các y/c thiết kế t/p xử lý of ht pm? Minh họa**

### **Câu 27: thế nào là ?**

- Thiết kế xử lý theo hướng chức năng : Top-Down

- Thiết kế xử lý theo hướng đối tượng : Bottom Up

Minh họa cho mỗi trường hợp

=====



Gợi ý:

. FO (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)

. OO (đối tượng) : Xuất phát từ các đối tượng cụ thể liên quan đến PM: Lập trình xử lý theo hướng đượng (Class Object = Attributes & Operators || Behavior || Methods) --...-> cấu thành PM (Bottom-Up)

### **Câu 28: Thiết kế các hệ thống giao tiếp khác (giao diện) của hệ thống phần mềm:**

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

- Giới thiệu một số yêu cầu và nguyên tắc chung:

. Gắn liền với chức năng của PM

. Đảm bảo Không làm ảnh hưởng đến kết cấu PM hiện có

. Đảm bảo an toàn & bảo mật đối với PM

. Phù hợp với sự phát triển của KH CN (ko sd ...lạc hậu)

. Khái thác hiệu quả các chức năng of HT

### **Câu 29: Thiết kế hệ thống bảo mật và an toàn của hệ thống phần mềm:**

- Thế nào là bảo mật HTPM? minh họa.

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?

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

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

### **Câu 30: Các cấp độ bảo mật trong giao dịch phần mềm trên Internet? Thường dùng chữ ký số trong các trường hợp PM ntn? Đề xuất cách nhận diện nsd (tw các quyền) vd: nhận diện qua login, ...**

- 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 nsd hạn chế

+ Passport = chữ ký số = Số NSD rất lớn = Public key [PIN] & Private key dựa SHA / RSA (VD: user / facebook, gmail) || phí

⇒ Ký số:

+ bảo mật = tránh mạo danh / nặc danh

+ số user quá lớn

=====

d/x cách nhận diện nsd (tw các quyền) vd: nhận diện qua login, ...

**Câu 31: Phân quyền người sử dụng:**

- + Các cấp độ “quyền” thông dụng của người sử dụng?
- + Giới thiệu một số phương pháp nhận diện “quyền” của người sử dụng trên các phần mềm

Gợi ý:

+ Quyền:

. Xem (View)

. Modify = Update

. Xóa

. Create

. Delete

. Copy

. Move

. Admin

. Owner (sở hữu)

. Liệt kê (List = Browser)

...

+ 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 (tw) với chức năng của cá nhân đó, NV NS không thực hiện các hạch toán kế toán, nhân viên KT ko làm các NV NS.

**Câu 32: Thiết kế hệ thống trợ giúp (Help) của hệ thống phần mềm**

- Giới thiệu một số yêu cầu và nguyên tắc cơ bản về hệ thống trợ giúp của phần mềm?
- Giới thiệu một số hình thức hỗ trợ người sử dụng thông dụng

**GOI Ý:**

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

====

[1] Hạn chế NSD sai sót khi SD PM

. Hỗ trợ tại mỗi bước đảm bảo hạn chế thao tác sai ...: ToolTip

[2] trực quang (WYSIWYG)

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

-> NSD biết được vị trí làm việc của họ / PM: đã làm, sẽ làm gì tiếp theo -> hình dung tổng thể những việc họ sẽ làm / PM

[3] Trợ giúp phải triệt để

[4] Tập trung các chức năng & thao tác chính (không lạm dụng)

[5] Đảm bảo NSD khai thác sd ALL các chức năng mà PM có

. ....

=====

Hình thức:

+ Text file

+ Tooltip

+ Đồ họa

+ Sơ đồ cây

+ Hướng dẫn context = hướng dẫn theo 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: XÉT form Màn hình of pm ql dự án [tr 8 của đề cương ], theo Anh /Chị nhưng 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:**

#### **● 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 pm

. Ít dùng

#### **● Hotkeys nhất quán**

Vd1: các sp pm ms: thoát = Alt\_F4

Một số tr/h dùng Ctrl\_C (cancel) : ko nên, vì nhầm với Copy

Vd2: Ctrl\_C / Ctrl\_V / Ctrl\_X

Vd3: Ctrl\_Z / Ctrl\_Y

...Giúp nsd dễ nhớ / ko phải học nhiều / ít nhầm lẫn

#### **● Hotkeys thống nhất ngôn ngữ giúp nsd dễ nhớ**

VD: Eng / Việt : Ctrl\_T (Tim) Ctrl\_F

KO 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 họa 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 (Setup 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)