

**Thiết kế giao diện**

# people

recent



all



"Heading to the gym now"



Amy Alberts



Dave Alexander



Luis Alverca

what's new

Victor Stehmann

2

Willows Live  
Updated his status 23 minutes ago  
"Taking the grandkids to the zoo."

Shirley Bassli

Facebook  
Liked on Luis Alverca's wall  
"Thank you for showing me around yesterday. I really like the new downtown area and that wine bar."

Alicia Kerry

5

Facebook  
Liked on Alicia Thomber's wall  
"Me waves today! I cannot believe many people were at the beach! See you again sometime. Next week."





Wed Apr 28, 2:00 PM ubuntu



Favorites



Files



Accessories



Games



Graphics



Internet



Office



Sound



System



Firefox Web  
Browser



Evolution Mail  
and Calendar



Cheese Webcam  
Booth



Empathy IM  
Client



Ubuntu Software  
Center



Install  
Ubuntu-Netbook  
10.04

SOFTPEDIA™

[www.softpedia.com](http://www.softpedia.com)





# Mục tiêu

---

- Gợi ý một số **nguyên lý thiết kế** tổng quát cho việc thiết kế giao diện người dùng
- Giải thích các **kiểu tương tác** khác nhau và ứng dụng của chúng
- Giải thích khi nào dùng biểu diễn thông tin dạng đồ họa và dạng text
- Giải thích các hoạt động chính trong quy trình thiết kế giao diện người dùng
- Giới thiệu các thuộc tính dùng được (**usability**) và các cách đánh giá hệ thống



# Các chủ đề

---

- Các vấn đề khi thiết kế
- Quy trình thiết kế UI
- Phân tích người dùng
- User interface prototyping
- Đánh giá giao diện

# Giao diện người dùng

---

- Nên được thiết kế cho phù hợp với kĩ năng, kinh nghiệm và sự trông đợi của người dùng tương lai của hệ thống.
- Người dùng hệ thống thường đánh giá một hệ thống theo giao diện thay vì chức năng.
- Một giao diện thiết kế tồi
  - có thể dẫn đến việc một người dùng phạm những lỗi nghiêm trọng.
  - là lí do vì sao nhiều hệ thống phần mềm không bao giờ được đem ra sử dụng.



# Nhân tố con người trong thiết kế giao diện

---

7

- Trí nhớ ngắn hạn có hạn
  - Người ta có thể nhớ nhanh khoảng 7 thông tin.
    - Nếu bạn đưa ra nhiều hơn như vậy, người ta sẽ dễ nhầm lẫn.
- Nhầm lẫn là chuyện thường
  - Khi người dùng nhầm lẫn và hệ thống chạy sai, những cảnh báo và thông báo không đúng mức có thể làm người dùng căng thẳng hơn và càng dễ nhầm lẫn hơn.
- Mỗi người có năng lực cao thấp khác nhau
  - Người thiết kế không nên chỉ thiết kế cho năng lực của chính mình.
- Mỗi người hợp với những kiểu tương tác khác nhau
  - Người này thích hình ảnh, người khác thích chữ.

# Các nguyên tắc thiết kế UI

---

- Thiết kế UI phải xét đến nhu cầu, kinh nghiệm và năng lực của người dùng hệ thống.
- Người thiết kế cần
  - nhận thức được các hạn chế về vật lý và tâm lý của con người
    - Ví dụ giới hạn của trí nhớ ngắn hạn
  - nhận thức rằng con người ai cũng có thể nhầm lẫn.
- Các nguyên tắc thiết kế UI đóng vai trò nền tảng cho các thiết kế giao diện
  - Tuy rằng không phải tất cả các nguyên tắc đều áp dụng được cho tất cả các thiết kế.

# Các nguyên tắc thiết kế UI

Nguyên tắc	Mô tả
<b>Quen thuộc với người dùng</b>	Giao diện nên dùng các thuật ngữ và khái niệm rút ra từ kinh nghiệm của những người sẽ dùng hệ thống nhiều nhất.
<b>Nhất quán</b>	giao diện cần nhất quán sao cho các thao tác gần giống nhau có thể được kích hoạt theo cùng kiểu.
<b>ngạc nhiên tối thiểu</b>	Người dùng không bao giờ bị bất ngờ về hành vi của hệ thống.
<b>khôi phục được</b>	Giao diện nên có các cơ chế cho phép người dùng khôi phục lại tình trạng hoạt động bình thường sau khi gặp lỗi.
<b>hướng dẫn người dùng</b>	Giao diện nên có phản hồi có nghĩa khi xảy ra lỗi và cung cấp các tiện ích trợ giúp theo ngữ cảnh.
<b>người dùng đa dạng</b>	Giao diện nên cung cấp các tiện ích tương tác thích hợp cho các loại người dùng hệ thống khác nhau.

# Quiz

---

- Câu hỏi 1: Đối với nguyên lý "Sự rõ ràng", điều gì quan trọng nhất khi thiết kế giao diện?**
  - A. Sử dụng nhiều màu sắc đa dạng.
  - B. Tăng cường hiệu ứng đồ họa.
  - C. Làm cho thông tin quan trọng nổi bật và dễ hiểu.
  - D. Sử dụng font chữ với kích thước nhỏ.
- Câu hỏi 2: Nguyên lý "Tính thẩm mỹ" trong thiết kế giao diện đề cập đến điều gì?**
  - A. Chỉ sử dụng màu đen và trắng.
  - B. Tạo ra giao diện đồ họa phức tạp.
  - C. Đảm bảo giao diện đẹp mắt và hấp dẫn người dùng.
  - D. Tối giản hóa mọi yếu tố trên giao diện.
- Câu hỏi 3: Nguyên lý "Đồng nhất" là gì trong thiết kế giao diện?**
  - A. Sự thay đổi liên tục để tạo nổi bật.
  - B. Sự đa dạng trong màu sắc và hình ảnh.
  - C. Sự nhất quán và thống nhất trong cả giao diện.
  - D. Sự phức tạp của các yếu tố trực quan.
- Câu hỏi 4: Trong nguyên lý "Tương tác người dùng", điều gì quan trọng nhất?**
  - A. Sự đẹp mắt của giao diện.
  - B. Khả năng tương tác dễ dàng và hiệu quả.
  - C. Số lượng thông tin trên mỗi trang.
  - D. Sử dụng nhiều hiệu ứng đặc biệt.
- Câu hỏi 5: Nguyên lý "Dễ sử dụng" trong thiết kế giao diện làm gì?**
  - A. Tạo ra giao diện phức tạp để thách thức người dùng.
  - B. Đảm bảo rằng người dùng có thể dễ dàng hiểu và sử dụng giao diện.
  - C. Sử dụng nhiều đối tượng đa phương tiện.
  - D. Tăng cường tính tương tác động đối với người dùng.

# Design principles

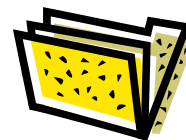
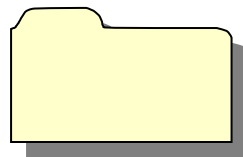
---

- Recoverability
  - The system should provide some resilience to user errors and allow the user to recover from errors.
    - Undo, confirmation of destructive actions, 'soft' deletes, etc.
- User guidance
  - Some user guidance such as help systems, on-line manuals, etc. should be supplied
- User diversity
  - Interaction facilities for different types of user should be supported.
    - For example, some users have seeing difficulties and so larger text should be available

# Các vấn đề khi thiết kế UI

---

- Hai vấn đề cần xem xét
  - Người dùng cung cấp thông tin cho hệ thống bằng cách nào?
  - Hệ thống nên trình bày thông tin (output) cho người dùng như thế nào?



# Các kiểu tương tác

- Thao tác trực tiếp – Direct manipulation
- Chọn lựa bằng menu – Menu selection
- Điền form – Form fill-in
- Dòng lệnh – Command language
- Ngôn ngữ tự nhiên – a Natural language



**WolframAlpha**™ computational knowledge engine

Who is Vo Nguyen Giap?

Input interpretation: **Võ Nguyên Giáp (military)** *Mathematica form*

Basic information:

full name	Võ Nguyên Giáp
date of birth	August 25, 1911 (age: 98 years)
place of birth	Vietnam

Timeline:

Võ Nguyên Giáp

1900 1925 1950 1975 2000

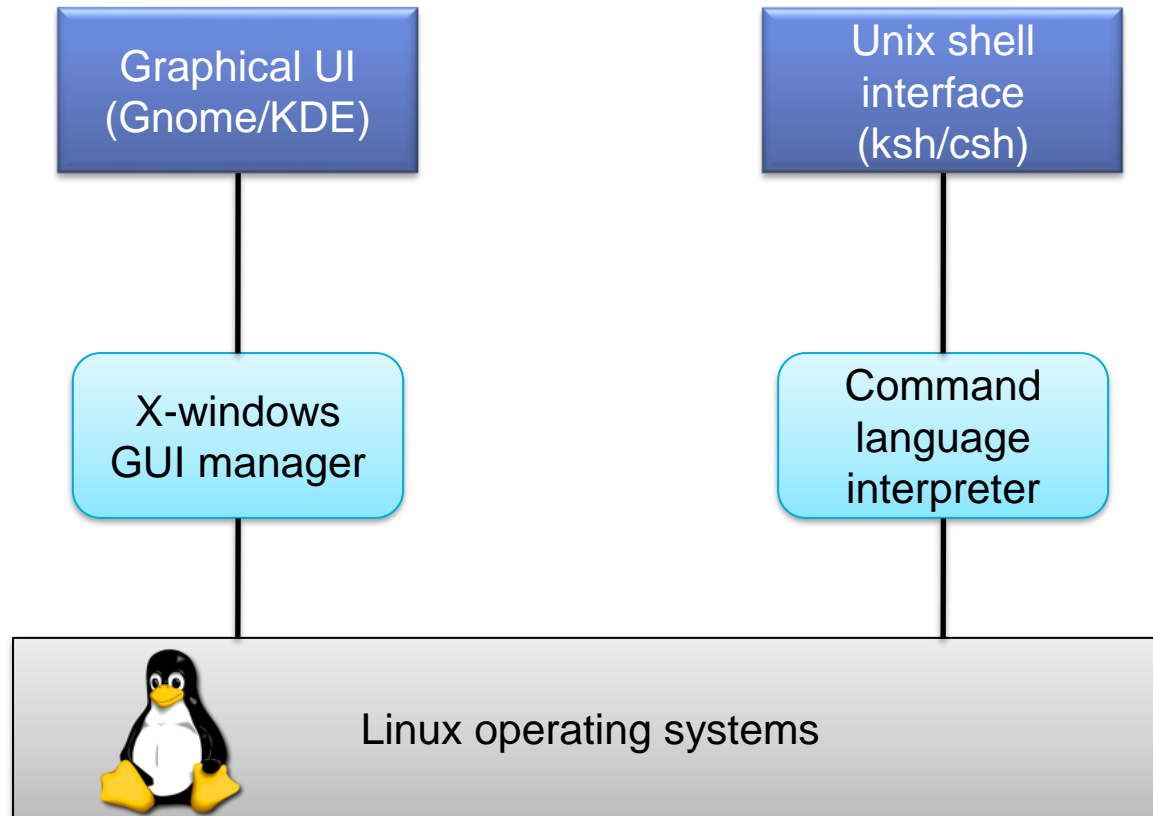


# Các kiểu tương tác

Kiểu tương tác	Ưu điểm chính	Nhược điểm chính	Ví dụ ứng dụng
<b>Direct manipulation</b>	Tương tác nhanh và trực quan Dễ học	Có thể khó cài đặt. Chỉ thích hợp khi có ẩn dụ hình ảnh cho các tác vụ và đối tượng.	Trò chơi điện tử Các hệ thống dùng drag-n-drop
<b>Menu selection</b>	Tránh lỗi người dùng Không phải gõ nhiều	Chậm chạp đối với người dùng nhiều kinh nghiệm. Có thể phức tạp nếu có nhiều lựa chọn menu.	Đa số các hệ thống thông dụng
<b>Form fill-in</b>	Nhập dữ liệu đơn giản Dễ học Kiểm tra được	Tốn không gian màn hình. Gây rắc rối khi các lựa chọn của người dùng không khớp với các trường của form.	Khai thuế, xử lý nợ cá nhân
<b>Command language</b>	mạnh và linh động	Khó học xử lý lỗi kém	hệ điều hành
<b>Natural language</b>	Đáp ứng được người dùng không chuyên Dễ mở rộng	Cần gõ nhiều. Các hệ thống hiểu ngôn ngữ tự nhiên không đáng tin cậy.	Information retrieval systems

# Multiple user interfaces

---



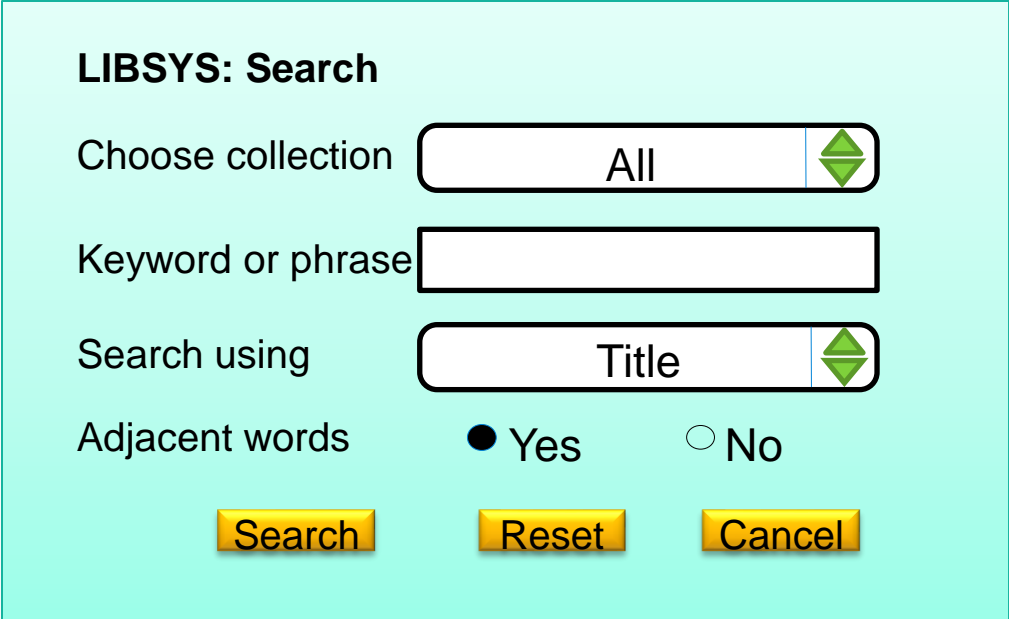
# LIBSYS interaction

---

- Tìm kiếm tài liệu
  - Người dùng cần có khả năng dùng các tiện ích tìm kiếm để tìm tài liệu mà họ cần.
- Yêu cầu tài liệu – Document request
  - Người dùng yêu cầu một tài liệu được chuyển tới máy tính của họ hoặc tới một server để in.

# Các giao diện dạng web

- Nhiều hệ thống web có các giao diện dựa trên các web form.
- Các trường của form có thể là menu, text input, radio button, v.v..
- Trong ví dụ LIBSYS, người dùng dùng menu để chọn nơi tìm kiếm và gõ chuỗi từ khóa vào trong một trường text.



**LIBSYS: Search**

Choose collection

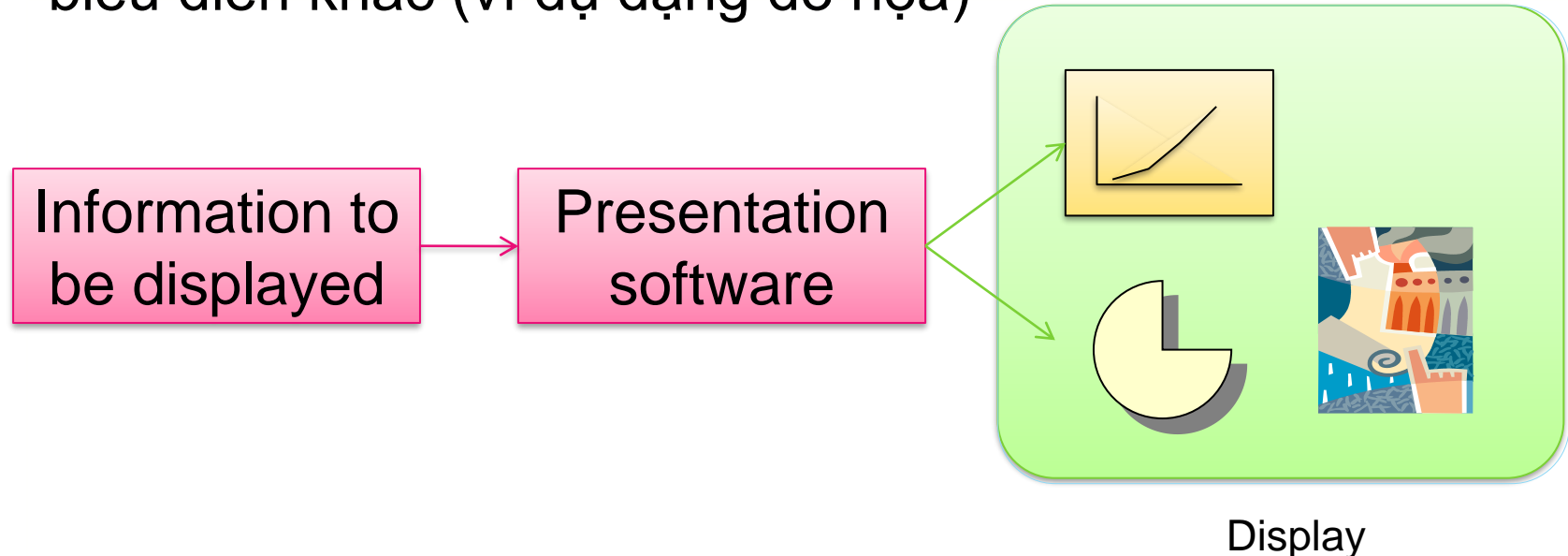
Keyword or phrase

Search using

Adjacent words ☒ Yes ☐ No

# Biểu diễn thông tin

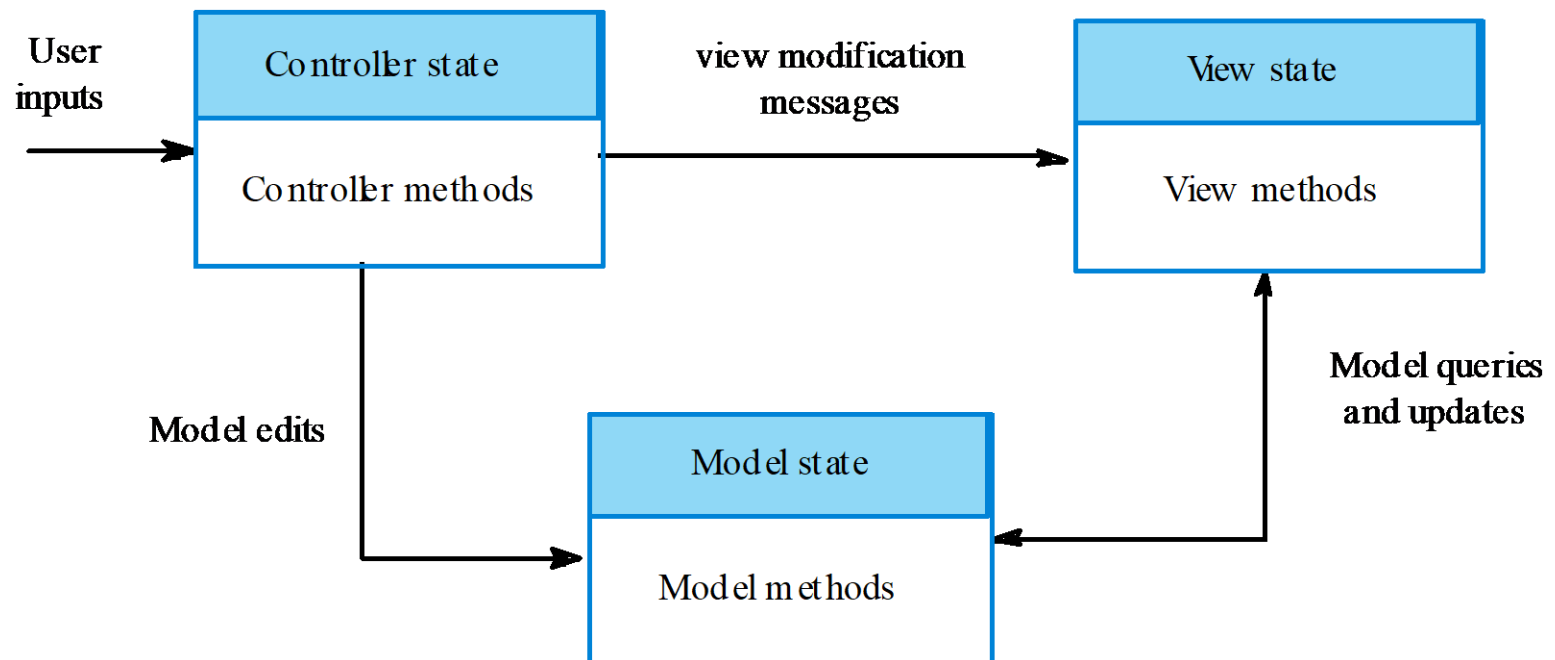
- Trình bày thông tin hệ thống như thế nào cho người dùng.
- Thông tin có thể được trình bày trực tiếp (ví dụ text trong một trình soạn thảo) hoặc được biến đổi thành một dạng biểu diễn khác (ví dụ dạng đồ họa)



# Model-view-controller

---

- Model-View-Controller là cách tiếp cận hỗ trợ nhiều kiểu biểu diễn dữ liệu



# Biểu diễn thông tin

---

- Thông tin tĩnh
  - Khởi tạo ở đầu session. Không thay đổi trong suốt session.
  - Có thể là số hoặc văn bản text.
- Thông tin động
  - Thay đổi trong session và phải được thông báo với người dùng.
  - Có thể là số hoặc văn bản text.



# Các nhân tố khi hiển thị thông tin

---

- Người dùng quan tâm đến con số chính xác hay mối tương quan giữa các số liệu?
- Các giá trị của thông tin thay đổi nhanh chậm ra sao? Có phải lập tức thông báo khi có thay đổi?
- Người dùng có phải hành động để phản ứng với một thay đổi?
- Có giao diện thao tác trực tiếp (direct manipulation) không?
- Thông tin dạng số hay text? Các giá trị tương đối có quan trọng không?

# Biểu diễn số hay biểu diễn tương tự

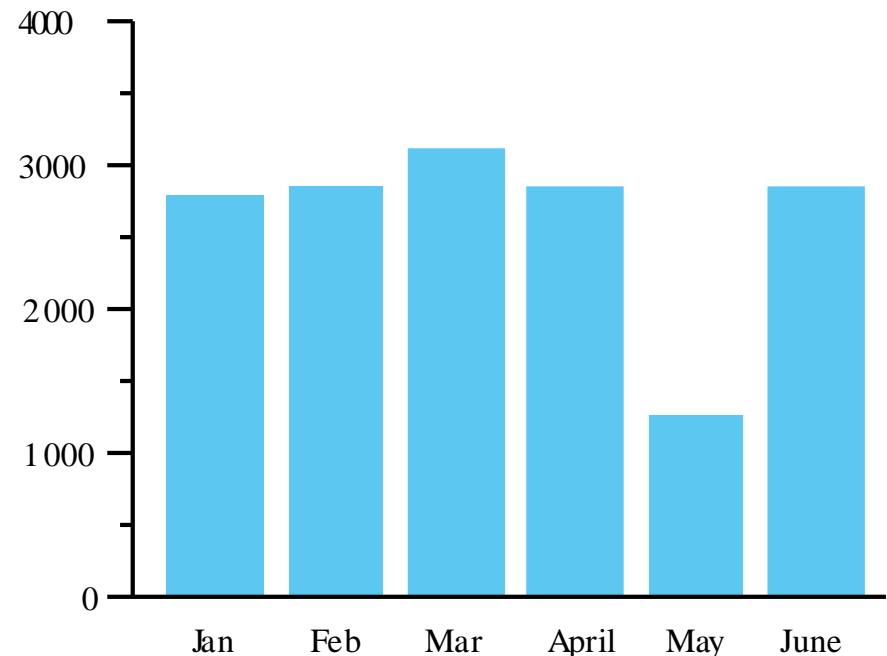
- Biểu diễn số - Digital presentation

- Ngắn gọn –  
chiếm ít không gian màn hình;
- Cho biết giá trị chính xác.

Jan	Feb	Mar	April	May	June
2 842	2 851	3 164	2 789	1 273	2 835

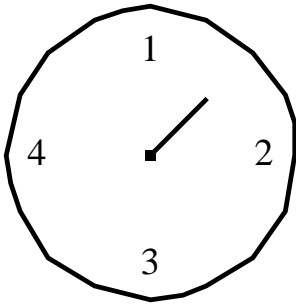
- Biểu diễn tương tự -  
Analogue presentation

- Nhanh chóng lấy được  
ấn tượng về một giá trị;
- Có thể biểu diễn các  
giá trị tương đối;
- Dễ thấy các  
giá trị dữ liệu đặc biệt.

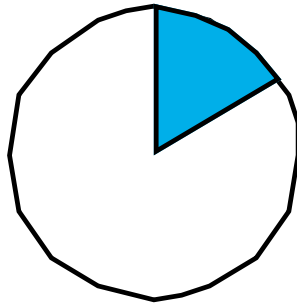


# Các phương pháp trình bày

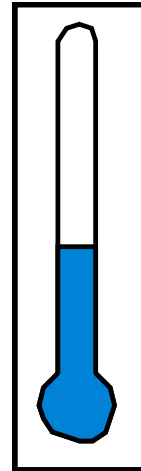
---



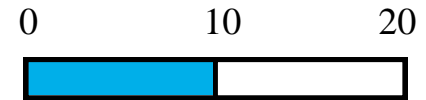
Dial with needle



Pie chart



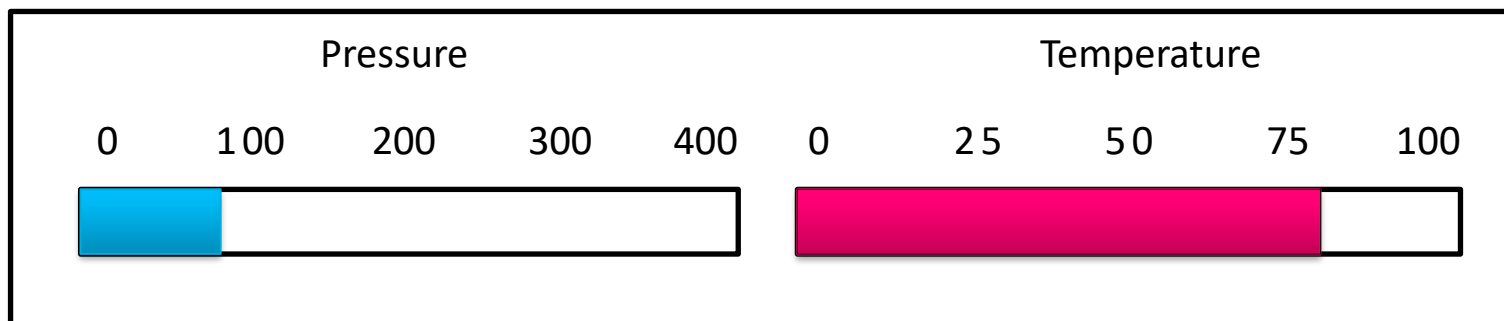
Thermometer



Horizontal bar

# Hiển thị các giá trị tương đối

---



# Data visualisation

---

- Các kĩ thuật hiển thị lượng lớn thông tin.
- Hình ảnh có thể cho thấy quan hệ giữa các thực thể và các xu hướng của dữ liệu.
- Các ví dụ:
  - Thông tin thời tiết thu thập từ nhiều nguồn;
  - Trạng thái của một mạng điện thoại dưới dạng một tập các nút được kết nối với nhau;
  - Một mô hình phân tử hiển thị dạng ba chiều;
  - Các trang web hiển thị dạng một cây hyperbol.

# Hiển thị màu

---

- Màu sắc bổ sung thêm một chiều cho một giao diện và giúp người dùng hiểu các cấu trúc thông tin phức tạp.
- Có thể dùng màu để highlight các thông tin đặc biệt.
- Các lỗi thường gặp trong việc dùng màu sắc trong thiết kế UI:
  - Dùng màu sắc để diễn đạt ý nghĩa;
  - Lạm dụng màu sắc trong trình bày.



# Hướng dẫn về dùng màu

---

- Hạn chế số màu và mức độ sắc sỡ.
- Dùng sự thay đổi màu để báo hiệu thay đổi trạng thái hệ thống.
- Dùng kí hiệu màu (color coding) để hỗ trợ công việc người dùng đang cố làm.
  - Highlight những điểm người dùng cần chú ý
- Dùng kí hiệu màu một cách cẩn trọng và nhất quán.
  - Nếu màu đỏ được dùng cho các thông báo lỗi không nên dùng màu đỏ cho các thông báo dạng khác để tránh người dùng nhầm lẫn các thông báo màu đỏ là thông báo lỗi.
- Cẩn thận về hiệu ứng cặp đôi của màu sắc.
  - Một số tổ hợp màu gây khó đọc. Ví dụ người ta không thể cùng lúc chú ý cả màu đỏ và màu xanh lam.



# Các thông báo lỗi

---

- Thiết kế thông báo lỗi là việc rất quan trọng
  - Các thông báo lỗi kém có thể dẫn đến việc người dùng không chấp nhận sử dụng một hệ thống.
- Các thông báo nên lịch sự, ngắn gọn, nhất quán và mang tính xây dựng.
- Nên xét đến background và kinh nghiệm của người dùng như là nhân tố quyết định khi thiết kế các thông báo lỗi.

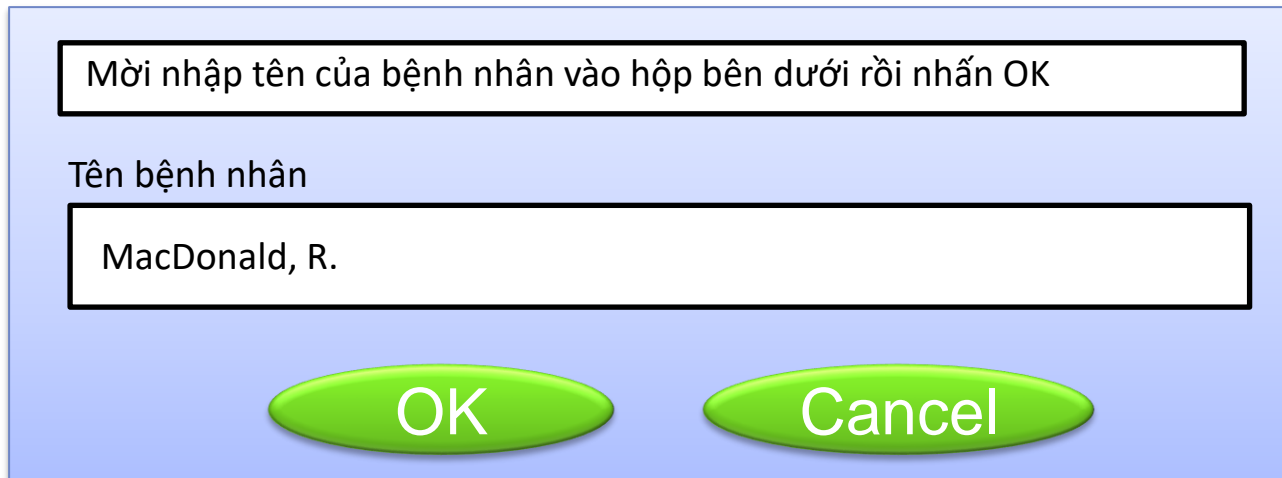
# Các nhân tố khi thiết kế nội dung các thông điệp

Nhân tố	Miêu tả
<b>Ngữ cảnh</b>	Các thông điệp do hệ thống tạo ra cần phản ánh ngữ cảnh sử dụng hiện tại bất cứ khi nào có thể. Hệ thống cần cố gắng nhận biết được người dùng đang làm gì và nên tạo các thông điệp có liên quan đến hoạt động hiện tại của họ.
<b>Kinh nghiệm</b>	Khi người dùng quen dần với một hệ thống, họ sẽ trở nên khó chịu bởi các thông điệp dài 'có ý nghĩa'. Tuy nhiên, những người mới dùng lại thấy những thông báo ngắn gọn khó hiểu. Bạn nên cung cấp cả hai loại thông điệp và để cho người dùng kiểm soát mức độ xúc tích của các thông báo.
<b>Trình độ kĩ năng</b>	Các thông điệp nên được điều chỉnh theo trình độ kĩ năng cũng như kinh nghiệm của người dùng. Thông điệp dành cho các loại người dùng khác nhau có thể được diễn đạt theo các cách khác nhau tùy theo bộ thuật ngữ quen thuộc với người đọc.
<b>Phong cách</b>	Các thông điệp nên mang tính tích cực hơn là tiêu cực, nên dùng thể chủ động hơn là bị động, không bao giờ nên có giọng xúc phạm hoặc cố pha trò.
<b>Văn hóa</b>	Nếu có thể, người thiết kế nên là người quen thuộc với văn hóa của đất nước nơi hệ thống sẽ được bán. Có những khác biệt văn hóa rõ rệt giữa châu Âu, châu Á và châu Mỹ. Một thông điệp phù hợp với nền văn hóa này có thể không chấp nhận được ở một nền văn hóa khác.

# Lỗi người dùng

---

- Giả sử một y tá đang cần tìm hồ sơ của một bệnh nhân nhưng lại gõ sai tên của bệnh nhân này.



Mời nhập tên của bệnh nhân vào hộp bên dưới rồi nhấn OK

Tên bệnh nhân

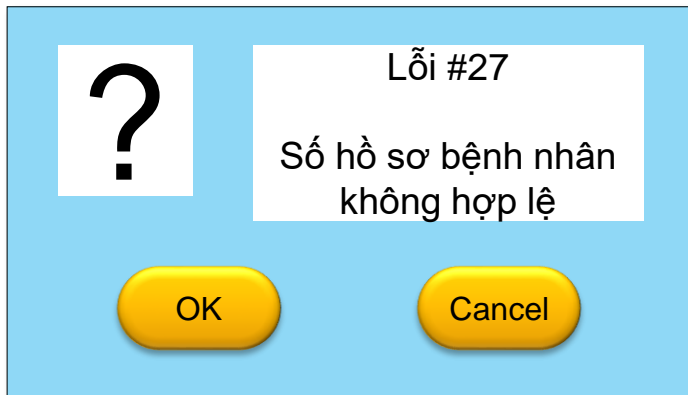
MacDonald, R.

OK Cancel

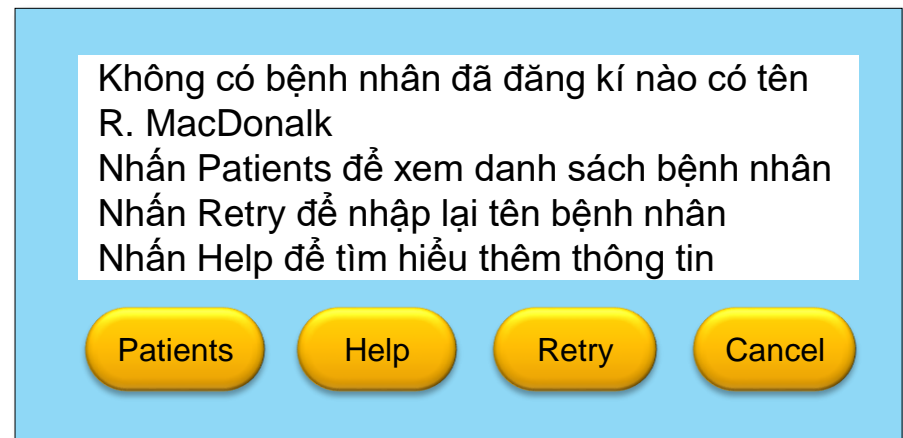
# Thiết kế thông điệp tốt và tồi

---

## Thông báo lỗi kiểu hệ thống



## Thông báo lỗi kiểu người dùng



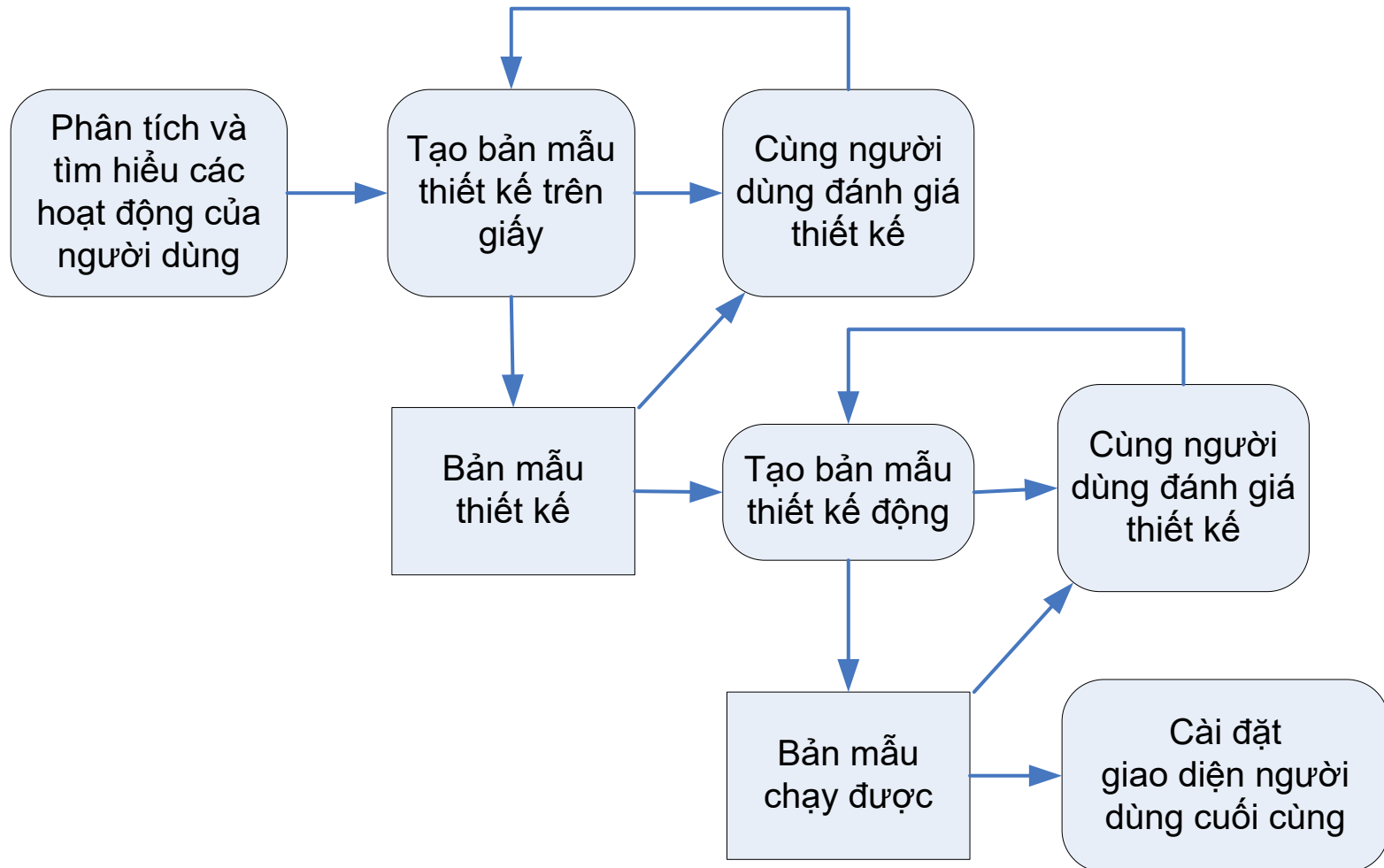
# Quy trình thiết kế UI

---

- Thiết kế UI là một quy trình lặp đi lặp lại với sự liên lạc chặt chẽ giữa người dùng và người thiết kế.
- Ba hoạt động chính trong quy trình:
  - **User analysis:** Tìm hiểu người dùng sẽ làm gì với hệ thống;
  - **System prototyping:** phát triển một loạt các bản mẫu để thử nghiệm;
  - **Interface evaluation:** thử nghiệm các bản mẫu cùng với người dùng.

# Quy trình thiết kế

---



# Phân tích người dùng

---

- Nếu bạn không hiểu người dùng muốn làm gì với một hệ thống, bạn khó có thể thiết kế một giao diện hiệu quả.
- Các phân tích người dùng phải được mô tả theo cách mà cả người dùng lẫn người thiết kế đều có thể hiểu được.
- Các kịch bản mà bạn dùng để miêu tả các tình huống sử dụng điển hình là một cách để mô tả các phân tích này.



# Kịch bản tương tác người dùng

---

Jane is a student of Religious Studies and is working on an essay on Indian architecture and how it has been influenced by religious practices. To help her understand this, she would like to access some pictures of details on notable buildings but can't find anything in her local library.

She approaches the subject librarian to discuss her needs and he suggests some **search terms** that might be used. He also suggests some libraries in New Delhi and London that might have this material so they log on to the library catalogues and do some searching using these terms. They find some source material and place a request for photocopies of the pictures with architectural detail to be posted directly to Jane.

# Các yêu cầu rút ra từ kịch bản

---

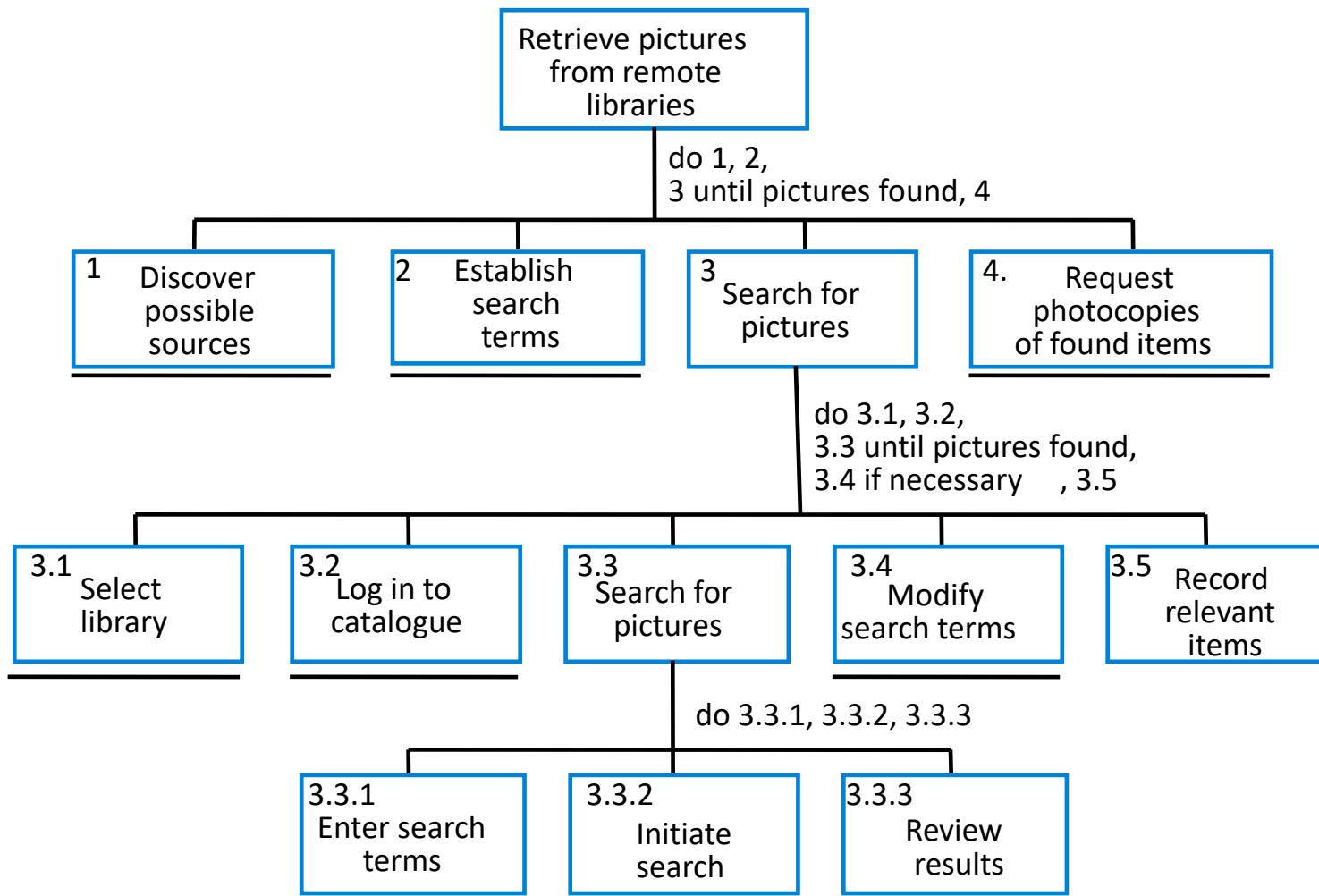
- Người dùng có thể không nhận ra các từ khóa tìm kiếm thích hợp, do đó cần có cách giúp đỡ họ chọn từ khóa.
- Người dùng phải có khả năng chọn collection cần tìm tại đó.
- Người dùng phải có khả năng thực hiện tìm kiếm và yêu cầu bản sao của tài liệu liên quan.

# Các kĩ thuật phân tích

---

- Phân tích tác vụ - Task analysis
  - Mô hình hóa các bước trong việc hoàn thành một tác vụ.
- Phỏng vấn và câu hỏi điều tra (questionnaire)
  - Hỏi người dùng về công việc của họ.
- Văn hóa
  - Quan sát người dùng khi họ đang làm việc.

# Phân tích tác vụ thành cây phân cấp



# Phỏng vấn

---

- Tổ chức các cuộc phỏng vấn có cấu trúc linh động dựa trên các câu hỏi mở.
- Khi đó người dùng có thể cung cấp các thông tin mà họ cho là quan trọng; không chỉ các thông tin mà bạn đã định thu thập.
- Phỏng vấn theo nhóm tạo điều kiện để người dùng thảo luận với nhau về công việc của họ.

# Văn hóa

---

- Một người quan sát người dùng khi họ đang làm việc và hỏi về công việc của họ (không theo kịch bản).
- Có giá trị vì nhiều tác vụ của người dùng mang tính trực quan và họ cảm thấy rất khó miêu tả và giải thích các tác vụ này.
- Giúp hiểu về vai trò của các ảnh hưởng xã hội và tổ chức đối với công việc.

# Hồ sơ tìm hiểu văn hóa

---

Một trạm điều khiển không lưu có một số 'bàn' điều khiển, trong đó các bàn điều khiển các vùng không gian cạnh nhau cũng được đặt cạnh nhau. Các chuyến bay trong một vùng được biểu diễn bằng các băng giấy lồng vào các giá gỗ theo thứ tự phản ánh vị trí của chúng trong vùng. Nếu không còn đủ chỗ trên giá (nghĩa là khi vùng không gian đó rất đông máy bay), những người điều khiển trải các băng giấy lên cái bàn viết đặt trước cái giá gỗ.

Khi chúng tôi quan sát những người điều khiển, chúng tôi nhận thấy cứ một lúc họ lại nhìn sang giá treo các băng giấy của vùng bên cạnh. Chúng tôi hỏi tại sao. Họ trả lời rằng, nếu bàn điều khiển bên cạnh phải trải giấy lên bàn, điều đó có nghĩa sẽ có nhiều chuyến bay bay vào vùng của họ. Vì vậy, họ sẽ cố gắng tăng tốc các chuyến bay trong vùng của mình để 'lấy chỗ' cho các chuyến bay đang đến.

# Rút ra từ tìm hiểu văn hóa

---

- Những người điều khiển phải nhìn thấy tất cả các chuyến bay trong một vùng. Do đó, phải tránh kiểu hiển thị với thanh kéo mà trong đó các chuyến bay trên đỉnh và dưới đáy màn hình có thể biến mất.
- Giao diện phải có cách nào đó để báo cho người điều khiển về số chuyến bay tại các vùng giáp ranh để họ có thể lập kế hoạch cho công việc của mình.



# Bản mẫu UI

---

- Mục tiêu của bản mẫu là cho phép người dùng có được trải nghiệm trực tiếp đối với giao diện.
- Nếu không có trải nghiệm trực tiếp, không thể đánh giá khả năng sử dụng một giao diện.
- Quy trình làm bản mẫu có thể có 2 bước:
  - Lúc đầu, có thể dùng bản mẫu trên giấy;
  - Sau đó tinh chỉnh thiết kế, và phát triển các bản mẫu tự động hóa với độ phức tạp ngày càng tăng.

# Bản mẫu trên giấy

---

- Dùng các bản vẽ nháp về UI để duyệt qua các kịch bản.
- Dùng storyboard để trình bày một chuỗi tương tác với hệ thống.
- Bản mẫu giấy là cách thu thập phản ứng của người dùng đối với một đề xuất thiết kế.

# Các kĩ thuật làm bản mẫu

---

- Script-driven prototyping
  - Viết một loạt các script và screen bằng các công cụ như Macromedia Director.
  - Khi người dùng tương tác, screen sẽ chuyển sang hiển thị màn hình tiếp theo.
- Visual programming
  - Dùng một ngôn ngữ chuyên dụng cho phát triển nhanh như Visual Basic.
- Internet-based prototyping
  - Dùng một trình duyệt web browser và các script kèm theo.

# Mock-up vs Wireframe vs Prototype

---

- <https://www.youtube.com/watch?v=o8WUsD6HENw> (Mock-up vs Wireframe vs Prototype - Are these not the same thing?)



## Prototype/Product Assessment Rubric

<b>Program Utility</b> <input type="checkbox"/> Solves a real world <b>problem</b> <input type="checkbox"/> <b>Creative</b> , original solution to the problem not currently available	Great, I want to buy it once it is finished	Creative and original	Resembles other programs that currently solve the same problem	Unable to figure out why someone would want to use the program
<b>Appropriateness to target user</b> <input type="checkbox"/> Prototype uses terms and symbols appropriate to target user <input type="checkbox"/> Look and feel appropriate to target user <input type="checkbox"/> Usability appropriate to target user	Perfect for target user	Designed for a well defined target user	Prototype designed without well-defined target user in mind	Designed for a computer science male, 20ish, computer science major
<b>Prototype's Interaction Usability</b> <input type="checkbox"/> User interface uses a simple and natural interaction style <input type="checkbox"/> User interface minimizes need to memorize	Intuitive, easy to learn, and easy to use	Easy to use once you learn it	Hard to use	Hard to figure out how to even get started using the prototype
<b>Completeness: Breadth of Interface</b> <input type="checkbox"/> Multiple features present in the interface (even if they are faked) <input type="checkbox"/> Prototype features go beyond a minimal implementation <input type="checkbox"/> Well thought out feature set	WOW – you provided a complete feature set in a horizontal prototype– with no extra fluff!	Your project was <i>too</i> easy to stretch your imagination but your feature set is complete	Your prototype lacks one or two important features that it should provide	Too much still remain to be done to evaluate the usefulness of the GUI
<b>Completeness: Depth of Interface</b> <input type="checkbox"/> All program features appear to be complete (even if they are faked) <input type="checkbox"/> No bugs apparent during demonstration	Everything needed in a vertical prototype is included	The working of some of the minor features is unclear	The working of some of the important features is unclear	Multiple features still do <b>not</b> exist in the prototype
<b>Sophistication of Interface</b> <input type="checkbox"/> Hard problem tackled with non-trivial solution <input type="checkbox"/> Non-interface features <i>faked</i> effectively	You took on a hard problem and had great ideas for solving it	Medium problem difficulty	Routine solution that should have been re-thought to provide more interesting challenges	Looks like a typical 2-week student project
<b>Prototype's Graphical Design</b> <input type="checkbox"/> Prototype is visually appealing <input type="checkbox"/> Components are laid out sensibly on the screen <input type="checkbox"/> Prototype's appearance is appropriate to target user	Great use of colors, fonts, graphics, and layout	Pleasant looking, clean, well organized GUI	Boring <i>or</i> Overly cluttered interface	User interface design seems inappropriate for problem area
<b>Creativity</b> <input type="checkbox"/> User interaction with prototype is creative <input type="checkbox"/> Feedback to user is creative <input type="checkbox"/> Interaction with prototype is intuitive and fun	You created a novel user interface that is natural to use	A few interesting tweaks to the conventional GUI interface	Some playful graphics and/or colors, but essentially a run of the mill windows GUI	Looks like a zillion other programs
<b>Prototype's Feedback to User</b> <input type="checkbox"/> Prototype provides clear feedback to user actions <input type="checkbox"/> Prototype provides clearly marked exit paths <input type="checkbox"/> Prototype provides error messages	Clear feedback, useful help, and error checking features are provided	Most actions provided clear feedback	Some actions lacked feedback	Many actions lacked feedback or error handling

Comments:

# Đánh giá UI

---

- Cần thực hiện một số đánh giá UI để đánh giá mức độ thích hợp.
- Đánh giá đầy đủ và toàn bộ thì quá đắt và không thực tế cho hầu hết các hệ thống.
- Về lí tưởng, một giao diện cần được đánh giá theo một đặc tả về usability. Tuy nhiên, người ta hiếm khi viết các đặc tả đó.

# Các thuộc tính usability

---

Thuộc tính	Miêu tả
Khả năng học	Người dùng mới cần bao lâu để có thể hoạt động hiệu quả với hệ thống?
Tốc độ vận hành	Tốc độ phản ứng của hệ thống có đáp ứng tốt công việc của người dùng?
Chịu lỗi	Mức độ dung thứ lỗi của hệ thống đối với lỗi người dùng.
Khả năng khôi phục	Khả năng hệ thống khôi phục từ lỗi của người dùng.
Tương thích	hệ thống gắn bó chặt chẽ với một kiểu làm việc đến đâu?

---

# Các kĩ thuật đánh giá đơn giản

---

- Câu hỏi điều tra để lấy phản hồi của người dùng.
- Quay video về việc sử dụng hệ thống rồi sau đó đánh giá nội dung.
- Cài các đoạn mã thu thập thông tin về các tiện ích được sử dụng và lỗi của người dùng.
- Phần mềm có chức năng thu thập phản hồi trực tuyến của người dùng.



# Tóm tắt

---

- Các nguyên lý thiết kế UI giúp thiết kế UI tốt.
- Các kiểu tương tác: direct manipulation, menu systems form fill-in, command languages và natural language.
- Nên dùng hiển thị đồ họa để biểu diễn các xu hướng và các giá trị tương đối. Dùng hiển thị số khi cần con số chính xác.
- Nên dùng màu sắc một cách hạn chế và nhất quán.

# Tóm tắt

---

- Quy trình thiết kế UI gồm: phân tích người dùng, làm bản mẫu hệ thống và đánh giá bản mẫu.
- Mục đích của phân tích người dùng là để người thiết kế nhận ra được cách hoạt động của người dùng.
- Quy trình làm bản mẫu UI nên bắt đầu từ bản mẫu giấy làm cơ sở cho bản mẫu tự động sau đó.
- Các mục tiêu của đánh giá UI là để thu thập phản hồi về cách cải tiến thiết kế giao diện và để đánh giá xem giao diện thỏa mãn đến đâu các yêu cầu về usability.

# Nguyên lý thiết kế website

---

- Đơn giản là tốt nhất
- Tính nhất quán
- Kiểu chữ & Khả năng đọc
- Bảng màu và hình ảnh
- Khả năng tương thích với thiết bị di động
- Website có tốc độ tải nhanh
- Điều hướng dễ dàng
- Giao tiếp hiệu quả

# Nguyên lý thiết kế website

- Đơn giản là tốt nhất

Trang web được thiết kế quá mức có thể không hiệu quả. Đặt quá nhiều yếu tố (elements) trên trang có thể dẫn đến việc khách truy cập mất tập trung vào mục đích chính của trang web của bạn. Sự đơn giản luôn đem đến sự hiệu quả khi thiết kế một website. Thiết kế trang web nhìn gọn gàng và mới mẻ không chỉ làm cho trang web hấp dẫn mà còn giúp người dùng điều hướng từ trang này sang trang khác một cách liền mạch.



# Nguyên lý thiết kế website

---

- **Tính nhất quán**

Tính nhất quán trong thiết kế trang web rất quan trọng, nó giúp website của bạn nhìn đẹp và chuyên nghiệp hơn. Hãy chú ý để thiết kế của bạn phù hợp với các thành phần (elements) thiết kế trong mỗi trang. Website của bạn cần thể hiện sự nhất quán ở phong chữ, kích thước, tiêu đề, tiêu đề phụ và kiểu nút CTA phải giống nhau trên toàn bộ trang web. Có nghĩa là bạn không dùng nhiều phong chữ, nhiều kích thước chữ, kiểu nút CTA trên website, điều này nhìn rối rắm và không thể hiện sự chuyên nghiệp.

Tìm ví dụ KHÔNG nhất quán trong thiết kế website.

# Nguyên lý thiết kế website

---

- Kiểu chữ & Khả năng đọc

Cần nhắc sử dụng phông chữ dễ đọc hơn. Các phông chữ sans-serif hiện đại như Arial, Helvetica... có thể được sử dụng cho phần nội dung văn bản. Một website có thiết kế đẹp cần tạo sự kết hợp thích hợp của các kiểu chữ cho từng thành phần thiết kế (elements) như tiêu đề, nội dung



# Nguyên lý thiết kế website

- Bảng màu và hình ảnh

Một giao diện website có sự kết hợp màu sắc hoàn hảo sẽ thu hút người dùng trong khi sự kết hợp kém có thể dẫn đến mất tập trung. Điều này đòi hỏi bạn phải chọn một bảng màu hoàn hảo cho trang web của mình để có thể tạo ra một cảm giác dễ chịu, để lại tác động tốt cho khách truy cập.

- Khả năng tương thích với thiết bị di động
- Website có tốc độ tải nhanh

Bạn có thể dùng các phần mềm chuyên dụng như Photoshop hoặc phần mềm nén ảnh online như: Tinipng, imagecompressor, img2go...để nén nhỏ ảnh lại. Ngoài ra cần tối ưu code, kết hợp mã thành một tệp CSS hoặc JavaScript trung tâm vì nó làm giảm các yêu cầu HTTP. Ngoài ra, nén HTML, JavaScript và CSS để tăng tốc độ tải trang web.



# Nguyên lý thiết kế website

---

- Điều hướng dễ dàng

Nghiên cứu cho thấy rằng khách truy cập ở lại nhiều thời gian hơn trên các trang web có điều hướng dễ dàng. Để điều hướng hiệu quả, bạn có thể cân nhắc tạo một hệ thống phân cấp trang hợp lý, sử dụng bread crumbs và thiết kế các nút có thể nhấp. Bạn nên tuân theo “quy tắc ba lần nhấp” để khách truy cập có thể nhận được thông tin cần thiết trong vòng tối đa ba lần nhấp chuột.

- Giao tiếp hiệu quả

Mục đích cuối cùng của khách truy cập là tìm kiếm thông tin và nếu trang web của bạn có thể giao tiếp hiệu quả với khách truy cập, rất có thể họ sẽ dành nhiều thời gian hơn trên trang web của bạn. Các thủ thuật có thể hiệu quả để thiết lập giao tiếp dễ dàng với khách truy cập là tổ chức thông tin bằng cách sử dụng tốt các tiêu đề chính và tiêu đề phụ, cắt bớt phần viết dài dòng và sử dụng gạch đầu dòng, thay vì các câu dài ngoằng.



# Nguyên lý thiết kế website

---

1. Trong thiết kế web, nguyên lý nào nhấn mạnh về sự dễ hiểu và dễ sử dụng cho người dùng? a) Nguyên lý màu sắc b) Nguyên lý tính linh hoạt c) Nguyên lý trải nghiệm người dùng (UX) d) Nguyên lý hiệu ứng
2. Đây là một trong những nguyên lý cơ bản của thiết kế web responsive? a) Tính đồng nhất (Consistency) b) Tính linh hoạt (Flexibility) c) Tính tương tác (Interactivity) d) Tính tự động thích ứng (Adaptability)
3. Nguyên lý "Mobile First" trong thiết kế web nhấn mạnh vào việc gì? a) Thiết kế trước cho các thiết bị di động trước khi thiết kế cho desktop b) Thiết kế trước cho desktop trước khi thiết kế cho các thiết bị di động c) Thiết kế chỉ cho các thiết bị di động d) Không liên quan đến việc thiết kế web
4. Trong thiết kế web, nguyên lý "F-pattern" ám chỉ điều gì? a) Một cách sắp xếp nội dung theo hình chữ F, tập trung vào việc đặt các phần quan trọng ở góc trên bên trái của trang b) Sự phân bố các phần tử theo hình chữ F trên trang web c) Tổ chức nội dung thành các phần theo hình chữ F d) Định dạng các tiêu đề và đoạn văn bản theo hình chữ F
5. Nguyên lý "Keep It Simple, Stupid" (KISS) trong thiết kế web nhấn mạnh vào điều gì? a) Đơn giản hóa giao diện và trải nghiệm người dùng b) Tạo ra giao diện phức tạp để thu hút người dùng c) Sử dụng nhiều hiệu ứng và đồ họa để làm cho trang web nổi bật d) Tăng cường tính tương tác trên trang web

# Nguyên lý thiết kế website

---

Tìm 8 ví dụ vi phạm 8 nguyên lý thiết kế.

- a. Mô tả bằng lời sự vi phạm.
- b. Link trang web vi phạm.
- c. Hình ảnh màn hình và khoanh đỏ vùng vi phạm.
- d. Đề xuất giải pháp khắc phục vi phạm.

Nộp link Google Doc [bit.ly/sub\\_activity](https://bit.ly/sub_activity)

# DrawIO Lab

---

- **Create wireframes and mockups in draw.io**
- <https://www.youtube.com/watch?v=UDZsjP4a7Tw>

# Mockup

---

- **Tiêu đề nhiệm vụ:** Tạo Mockup cho Ứng dụng Thi Trắc Nghiệm Tin Học Lớp 10
- **Mô tả nhiệm vụ:** Bạn được giao nhiệm vụ tạo một mockup cho một ứng dụng thi trắc nghiệm tin học dành cho học sinh lớp 10. Ứng dụng này sẽ cung cấp cho học sinh một cách tiện lợi để ôn tập và kiểm tra kiến thức về tin học thông qua các câu hỏi trắc nghiệm.

# Mockup

---

- **Yêu cầu:**

1. Tạo một giao diện người dùng đơn giản và trực quan cho ứng dụng, bao gồm các màn hình sau:
  1. Màn hình đăng nhập/đăng ký: Cho phép người dùng đăng nhập vào tài khoản của họ hoặc đăng ký một tài khoản mới.
  2. Màn hình chính: Hiển thị danh sách các bài thi có sẵn và các tùy chọn khác như "Ôn tập", "Lịch sử thi", "Thông tin cá nhân" và "Đăng xuất".
  3. Màn hình bài thi: Hiển thị các câu hỏi trắc nghiệm và các tùy chọn để chọn đáp án.
  4. Màn hình kết quả: Hiển thị kết quả của bài thi sau khi hoàn thành, bao gồm số câu đúng, số câu sai và điểm số tổng cộng.
2. Thiết kế các biểu tượng, nút bấm và giao diện người dùng phải dễ nhận biết, dễ sử dụng và thân thiện với người dùng lớp 10.
3. Cần nhắc đến việc sử dụng màu sắc và hình ảnh phù hợp để làm cho giao diện trở nên hấp dẫn và thu hút.

# Mockup

---

- **Lưu Ý:**
- Mockup chỉ cần là bản vẽ tĩnh, không cần có chức năng hoạt động.
- Tập trung vào việc biểu diễn cấu trúc và giao diện tổng quan của ứng dụng, không cần quá chi tiết vào các tính năng cụ thể.
- Đảm bảo rằng giao diện mockup là sáng tạo và phản ánh được mục đích và tính chất của ứng dụng thi trắc nghiệm tin học.
- **Ghi chú:**
- Sau khi hoàn thành, vui lòng chia sẻ mockup của bạn thông qua tệp hình ảnh hoặc bản vẽ để được đánh giá và cung cấp phản hồi.

# Mockup

---

- Câu hỏi:
  1. Bạn sẽ sử dụng màu sắc và hình ảnh như thế nào để làm cho giao diện ứng dụng trở nên hấp dẫn và thân thiện với học sinh lớp 10?
  2. Làm thế nào để bạn đảm bảo rằng giao diện đăng nhập/đăng ký của ứng dụng là dễ sử dụng và dễ hiểu cho người dùng?
  3. Tại sao việc tạo mockup là bước quan trọng trong quá trình phát triển ứng dụng?