

**ĐẠI HỌC QUỐC GIA TP.HCM**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**NHẬP MÔN**  
**CÔNG NGHỆ PHẦN MỀM**

**Giảng viên: TS. Nguyễn Thị Xuân Hương**

**Email: [tuyendtt@uit.edu.vn](mailto:tuyendtt@uit.edu.vn)**

# NỘI DUNG MÔN HỌC

- Tổng quan về Công nghệ phần mềm
- Xác định và mô hình hóa yêu cầu phần mềm
- **Thiết kế phần mềm: – Thiết kế GIAO DIỆN**
- Cài đặt phần mềm
- Kiểm thử và bảo trì
- Đồ án môn học

# 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 bước thiết kế giao diện ứng dụng

# Nội dung

1. Thiết kế UI
2. Các nguyên lý khi thiết kế UI
3. Các vấn đề khi thiết kế UI
4. **Quy trình thiết kế UI**
5. Phân tích người dùng
6. Bản mẫu giao diện người dùng (User interface prototyping)
7. Một số nguyên tắc khi thiết kế UI
8. Đánh giá giao diện
9. Thiết kế giao diện ứng dụng

# 1. Thiết kế UI

- **Gợi ý thiết kế 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.

# 1. Thiết kế UI

## ■ Nhân tố con người trong thiết kế giao diện:

- ◆ 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.

## 2. Các nguyên lý khi thiết kế UI

- **Các vấn đề khi 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 lý Các vấn đề khi thiết kế UI đóng vai trò nền tảng cho các thiết kế giao diện**
  - ◆ Tuy rằng không phải tất cả các nguyên lý đều áp dụng được cho tất cả các thiết kế.

## 2. Các nguyên lý khi thiết kế UI

Nguyên lý	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 ứng xử 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.



## 2. Các nguyên lý khi thiết kế UI

### ■ Sự quen thuộc của người dùng

- ◆ Giao diện phải dựa trên các thuật ngữ và khái niệm hướng tới người dùng hơn là khái niệm máy tính.
- ◆ Ví dụ: một hệ thống văn phòng nên sử dụng các khái niệm như thư từ, tài liệu, thư mục, v.v. thay vì thư mục, mã định danh tệp, v.v.

### ■ Tính nhất quán

- ◆ Hệ thống phải hiển thị mức độ nhất quán thích hợp.
  - Các lệnh và menu phải có cùng định dạng, dấu câu lệnh phải giống nhau, v.v.

### ■ Bất ngờ tối thiểu

- ◆ Nếu một lệnh hoạt động theo cách đã biết, người dùng sẽ có thể dự đoán hoạt động của các lệnh tương đương

## 2. Các nguyên lý khi thiết kế UI

### ■ Khả năng phục hồi

- ◆ Hệ thống phải cung cấp khả năng phục hồi nhất định đối với lỗi của người dùng và cho phép người dùng khôi phục sau lỗi.
  - Hoàn tác, xác nhận các hành động phá hoại, xóa 'mềm', v.v.

### ■ Hướng dẫn sử dụng

- ◆ Cần cung cấp một số hướng dẫn sử dụng như hệ thống trợ giúp, sách hướng dẫn trực tuyến, v.v.

### ■ Sự đa dạng của người dùng

- ◆ Cần hỗ trợ các phương tiện tương tác cho các loại người dùng khác nhau.
  - Ví dụ: một số người dùng gặp khó khăn khi nhìn thấy và vì vậy nên có sẵn văn bản lớn hơn

# 3. Các vấn đề khi thiết kế UI

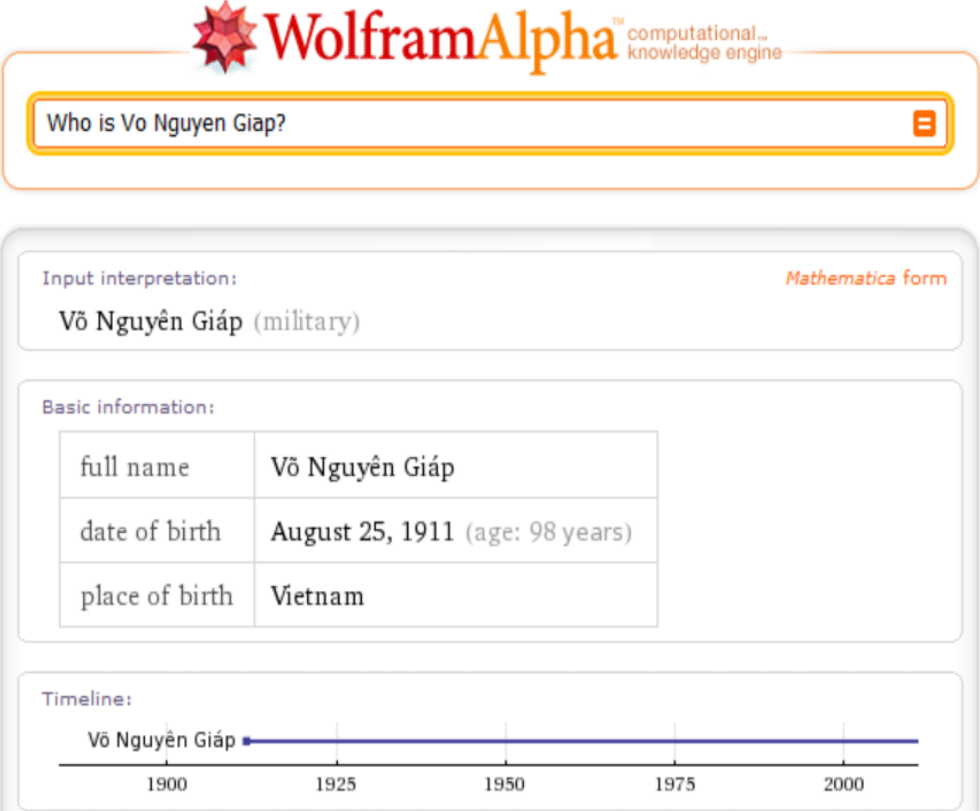
## ■ Hai vấn đề cần xem xét

- ◆ 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?

# 3. Các vấn đề khi thiết kế UI

## ■ 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



The screenshot displays the WolframAlpha interface. At the top, the logo 'WolframAlpha' is shown with the tagline 'computational knowledge engine'. Below the logo is a search bar containing the text 'Who is Vo Nguyen Giap?'. The results are organized into sections: 'Input interpretation' showing 'Võ Nguyễn Giáp (military)' with a 'Mathematica form' link; 'Basic information' presented as a table with details about Võ Nguyễn Giáp; and a 'Timeline' showing his life span from 1911 to 2000.

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

Timeline: Võ Nguyễn Giáp

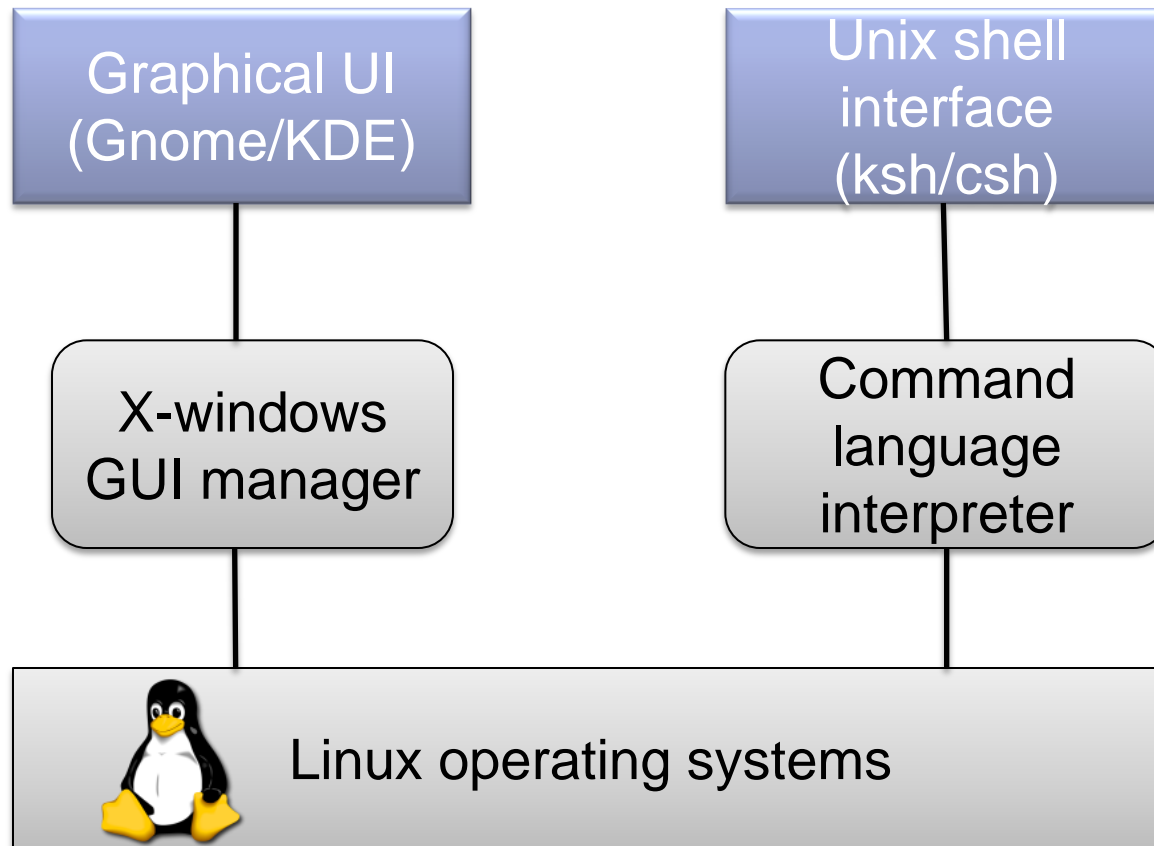
1900 1925 1950 1975 2000

### 3. Các vấn đề khi thiết kế UI – 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

### 3. Các vấn đề khi thiết kế UI

#### ■ Multiple user interfaces



# 3. Các vấn đề khi thiết kế UI

## ■ Ví dụ: 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.

### 3. Các vấn đề khi thiết kế UI

#### ■ Các giao diện dạng web/windows

- ◆ 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..

#### ■ Ví dụ LIBSYS, người dùng dùng menu để chọn nơi tìm kiếm và gõ chuỗi từ khóa vào trong một trường text.

The screenshot displays a web form titled "LIBSYS: Search" on a light blue background. It contains the following elements:

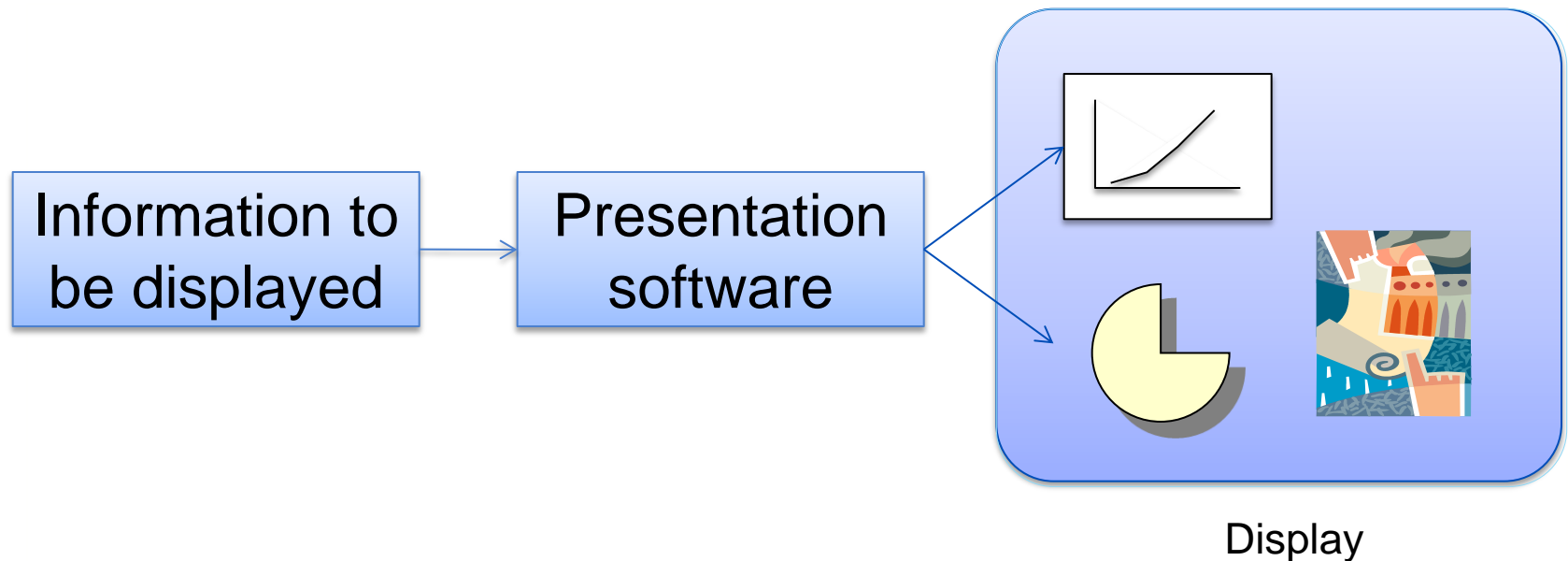
- Choose collection:** A dropdown menu with "All" selected and a blue/green arrow icon on the right.
- Keyword or phrase:** A text input field.
- Search using:** A dropdown menu with "Title" selected and a blue/green arrow icon on the right.
- Adjacent words:** Two radio buttons, "Yes" (selected) and "No".
- Buttons:** Three buttons at the bottom: "Search", "Reset", and "Cancel".



### 3. Các vấn đề khi thiết kế UI

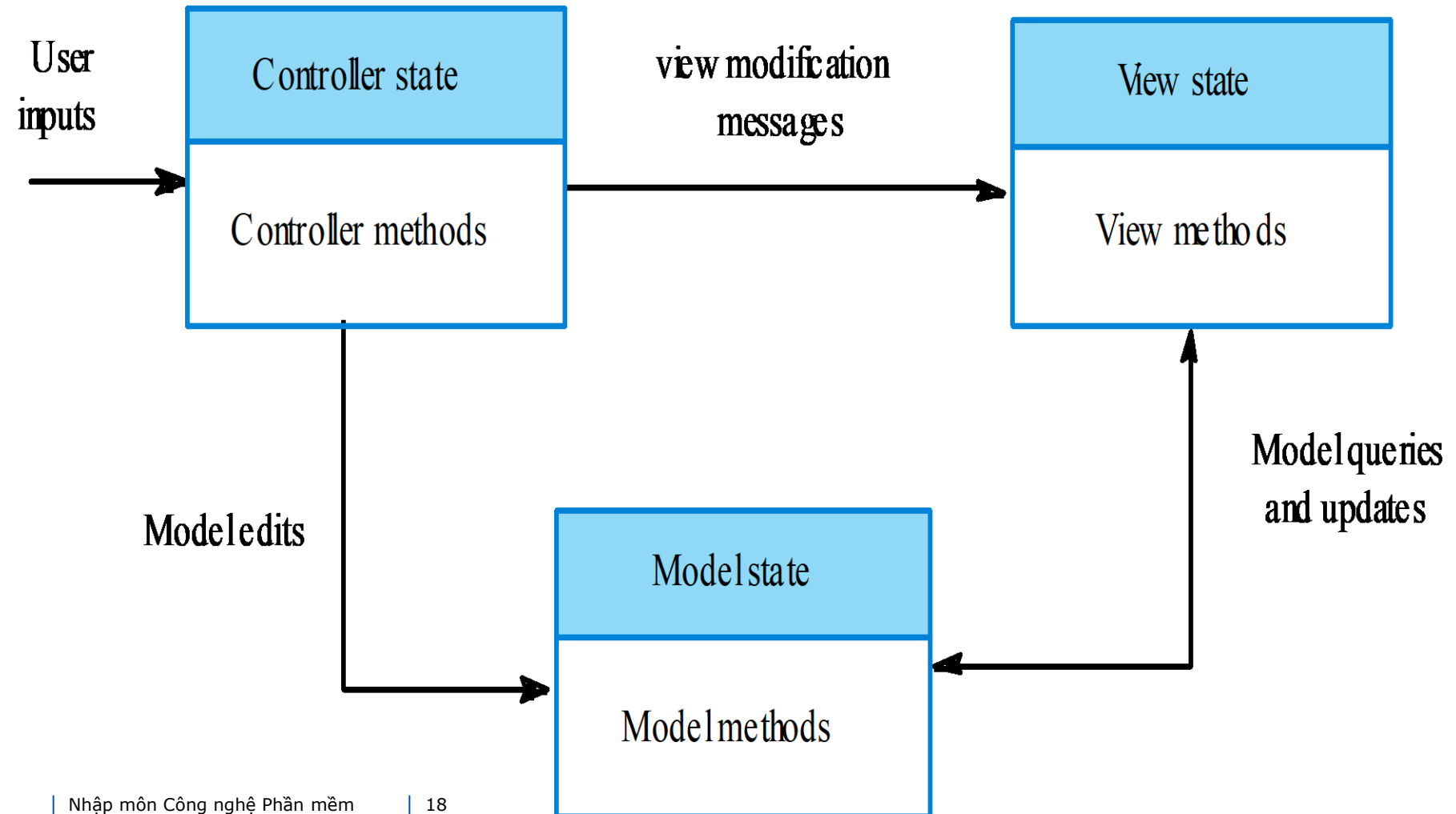
#### ■ Biểu diễn thông tin

- ◆ 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)



### 3. Các vấn đề khi thiết kế UI

#### ■ Biểu diễn thông tin: Model-view-controller



# 3. Các vấn đề khi thiết kế UI

## ■ 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

### 3. Các vấn đề khi thiết kế UI

#### ■ Các nhân tố khi hiển thị thông tin

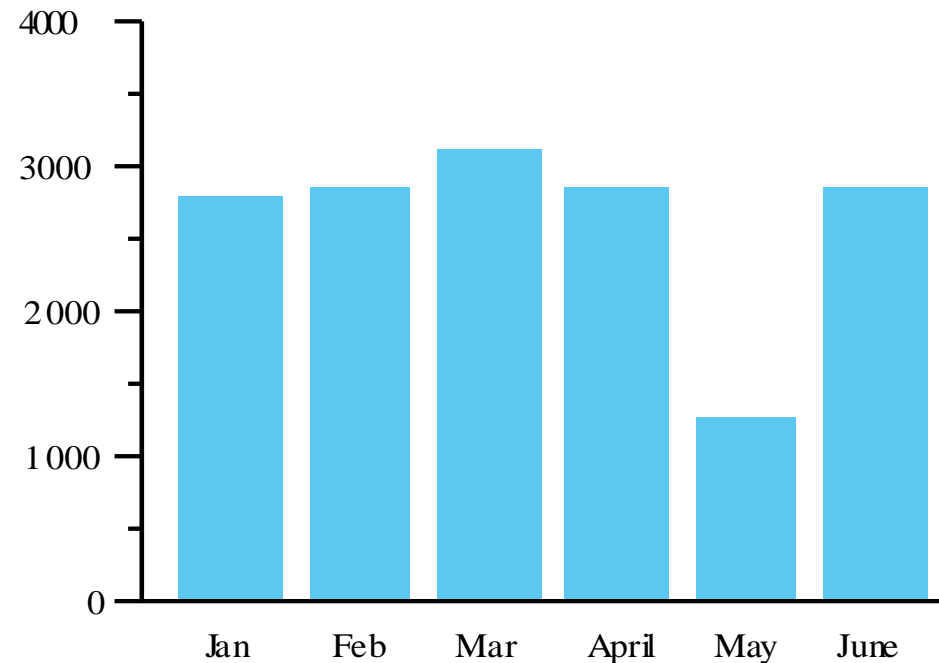
- ◆ Người dùng quan tâm đến con số chính xác hay mối tương quan giữa các số liệu?
- ◆ Các giá trị của thông tin thay đổi nhanh chậm ra sao? Có phải lập tức thông báo khi có thay đổi?
- ◆ Người dùng có phải hành động để phản ứng với một thay đổi?
- ◆ Có giao diện thao tác trực tiếp (direct manipulation)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

### 3. Các vấn đề khi thiết kế UI

#### ■ Các nhân tố khi hiển thị thông tin: Biểu diễn số hay biểu diễn tương tự

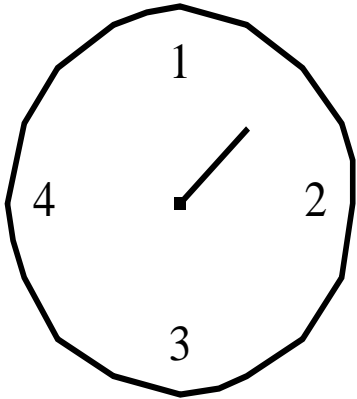
- ◆ **Biểu diễn số** - Digital presentation
  - Ngắn gọn – chiếm ít không gian màn hình;
  - Cho biết giá trị chính xác.
- ◆ **Biểu diễn tương tự** - Analogue presentation
  - Nhanh chóng lấy được ấn tượng về một giá trị; Có thể biểu diễn các giá trị tương đối;
  - Dễ thấy các giá trị dữ liệu đặc biệt.

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

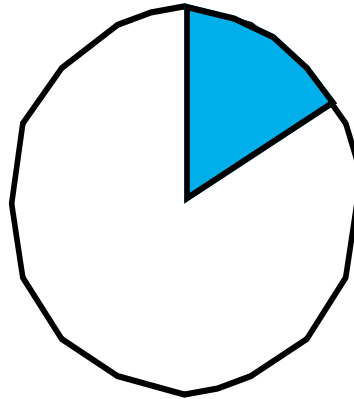


### 3. Các vấn đề khi thiết kế UI

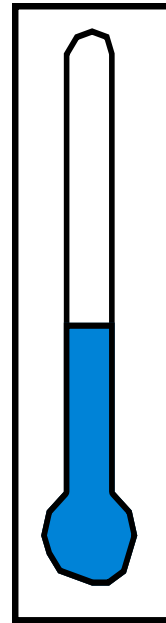
- Các nhân tố khi hiển thị thông tin: Các phương pháp trình bày



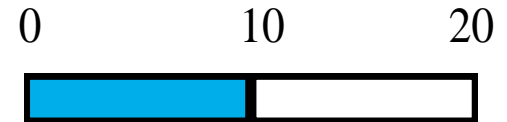
Dial with needle



Pie chart



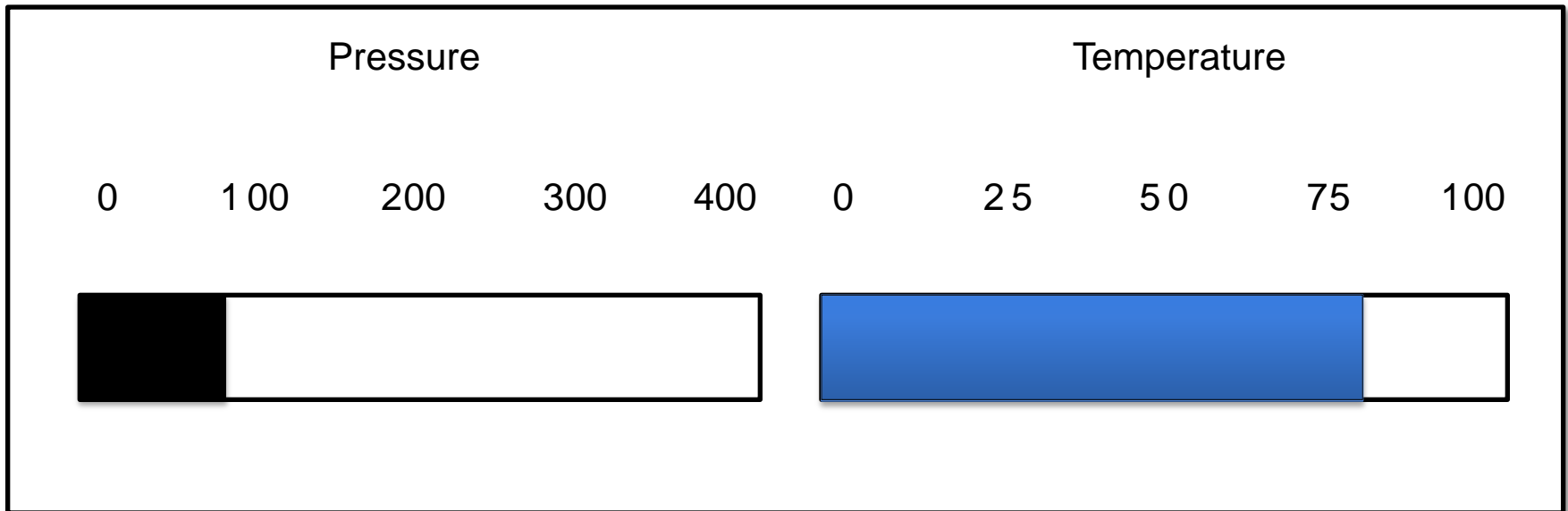
Thermometer



Horizontal bar

### 3. Các vấn đề khi thiết kế UI

- **Các nhân tố khi hiển thị thông tin:** Hiển thị các giá trị tương đối



# 3. Các vấn đề khi thiết kế UI

## ■ 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.



# 3. Các vấn đề khi thiết kế UI

## ■ Hiện 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 Các vấn đề khi thiết kế UI:
  - Dùng màu sắc để diễn đạt ý nghĩa;
  - Lạm dụng màu sắc trong trình bày.

### 3. Các vấn đề khi thiết kế UI

#### ■ **Hiện thị màu:** 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.

# 3. Các vấn đề khi thiết kế UI

## ■ Hiện thị văn bản và biểu tượng:

◆ Nhất quán trong toàn bộ hệ thống về:

- Cách trình bày: font chữ, màu sắc, vị trí các mục.
- Tên các nút điều khiển.
- Ý nghĩa biểu tượng.

# 3. Các vấn đề khi thiết kế UI

## ■ 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.

### 3. Các vấn đề khi thiết kế UI

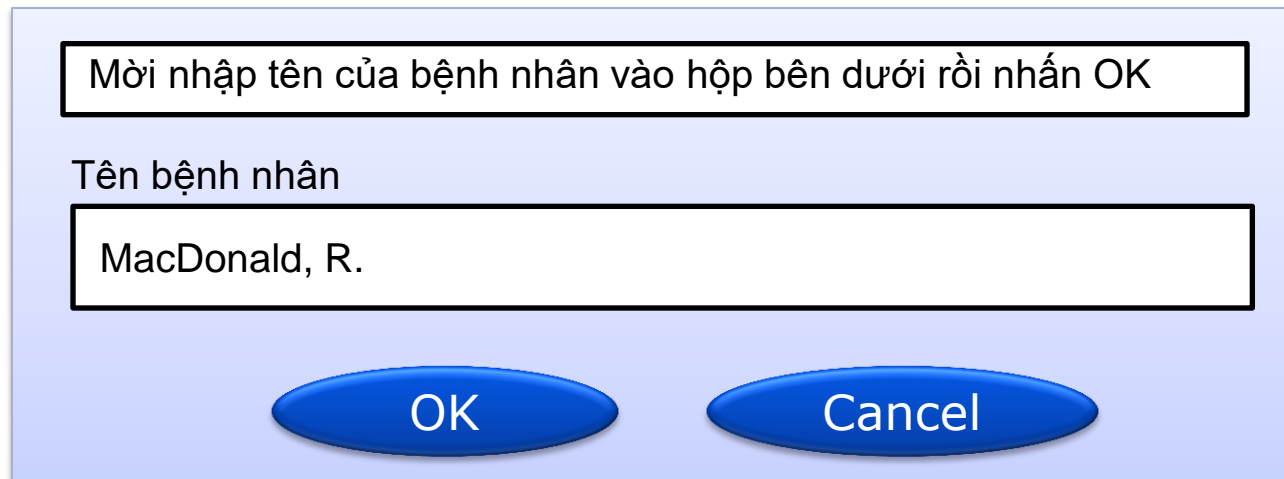
#### ■ Các nhân tố khi thiết kế nội dung các thông điệp

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.

### 3. Các vấn đề khi thiết kế UI

#### ■ Ví dụ: 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

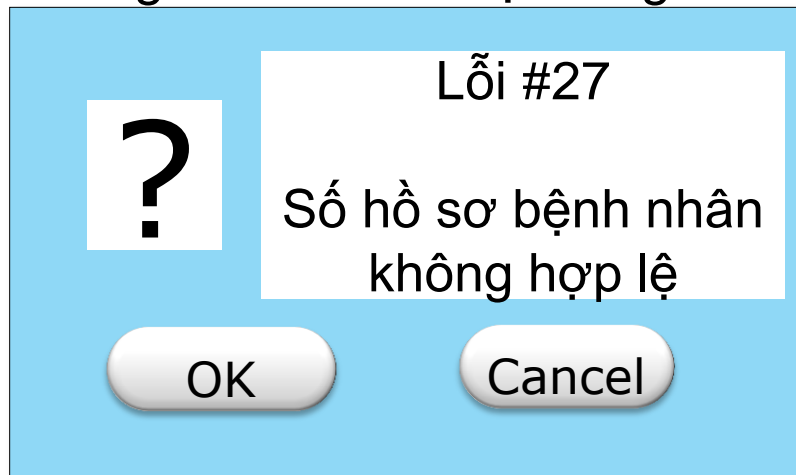
MacDonald, R.

OK Cancel

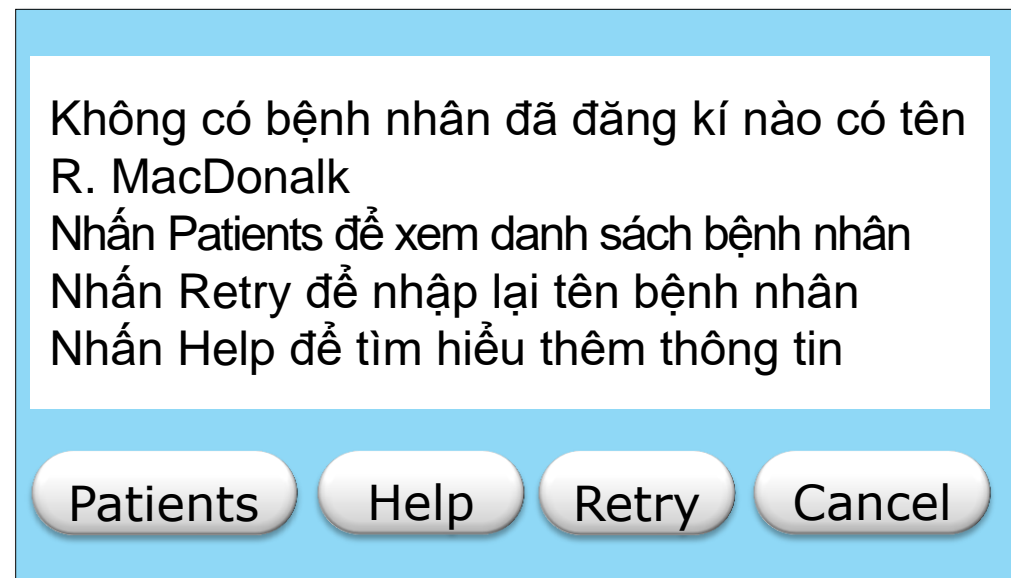
### 3. Các vấn đề khi thiết kế UI

#### ■ Ví dụ: lỗi người dùng:

Thông báo lỗi kiểu hệ thống



Thông báo lỗi kiểu người dùng

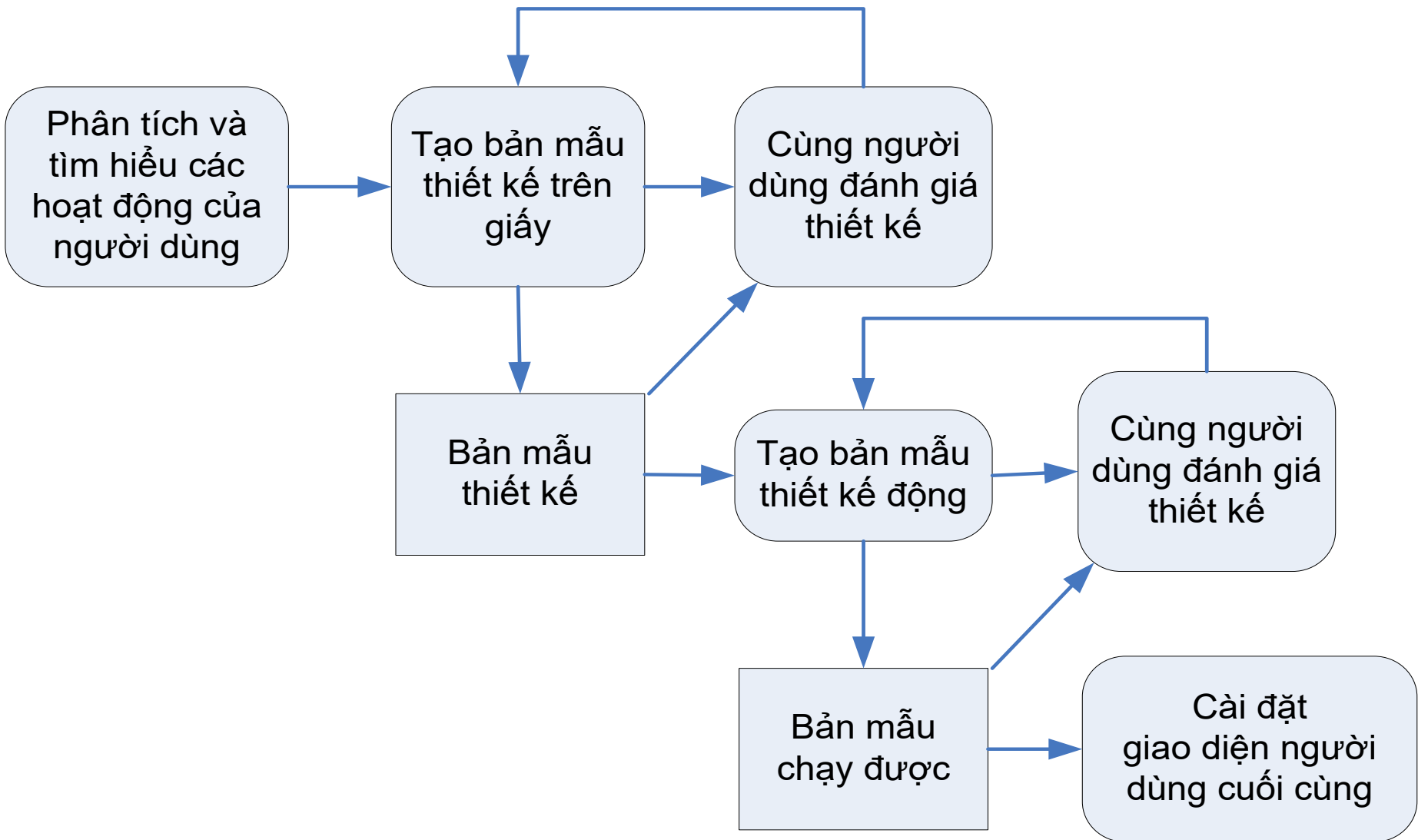


## 4. Quy trình thiết kế UI

- Các vấn đề khi 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.



## 4. Quy trình thiết kế UI



## 5. 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.

## 5. Phân tích người dùng

### ■ 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.

# 5. Phân tích người dùng

## ■ 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.

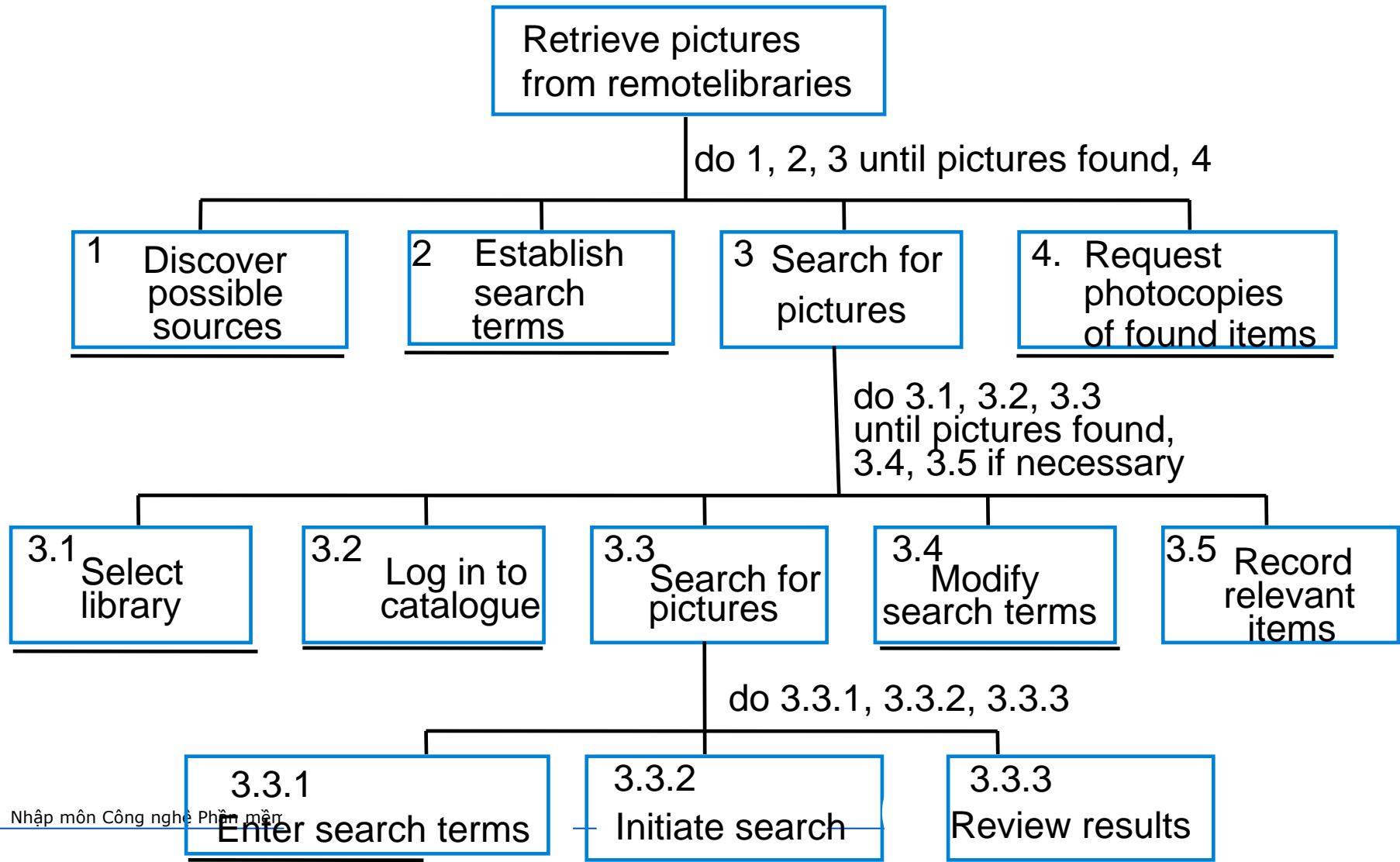
# 5. Phân tích người dùng

## ■ Các kỹ 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.

# 5. Phân tích người dùng

## ■ Phân tích tác vụ thành cây phân cấp



# 5. Phân tích người dùng

## ■ 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.

# 5. Phân tích người dùng

## ■ 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.



# 5. Phân tích người dùng

## ■ VD: 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.

## 5. Phân tích người dùng

### ■ VD: Hồ sơ tìm hiểu văn hóa

#### ⇒ 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.

## 6. 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.

# 6. Bản mẫu UI

## ■ 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.
- ◆ Công cụ hỗ trợ:
  - Có thể sử dụng các công cụ sau để hỗ trợ thiết kế giao diện: MS Word, Paint, Photoshop, Corel...
  - Dùng Crystal Report để thiết kế các báo biểu.

# 6. Bản mẫu UI

## ■ Các kỹ 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 studio.

### ◆ Internet-based prototyping

- Dùng một trình duyệt web browser và các script kèm theo.

# 7. Một số nguyên tắc khi thiết kế UI

- Tất cả màn hình phải có tên.
- Thiết kế phù hợp với đối tượng sử dụng.
- Dễ học, dễ nhớ, phù hợp với người mới sử dụng, đồng thời hỗ trợ các cách làm nhanh, làm tắt cho người sử dụng có kinh nghiệm.
- Chú ý môi trường triển khai ứng dụng cũng đòi hỏi những nguyên tắc thiết kế khác nhau do có sự khác nhau về tốc độ thực hiện các xử lý (web form, win form...).
- Số bước để đi đến một màn hình công việc chính phải  $\leq 3$

# 7. Một số nguyên tắc khi thiết kế UI

- Thứ tự trình bày trên màn hình phải phù hợp với văn hóa, thói quen của người sử dụng.
- Sắp xếp các control sao cho khoảng trống trong form được giảm tối đa.
- Kích thước các control phải cân đối, hài hoà, hợp lý.
- Chỉ trình bày những nội dung thật sự cần thiết, không trình bày quá nhiều thông tin trên một màn hình.
- Chọn font chữ rõ ràng, cỡ chữ phù hợp, dùng chữ IN/hoa đúng trường hợp.

# 7. Một số nguyên tắc khi thiết kế UI

## ■ Màu sắc:

- Màu sắc phải hài hòa, lưu ý kết hợp màu nền và màu chữ hợp lý.
- Nên dùng màu lạnh, chỉ dùng màu nóng khi cần gây chú ý (hoặc có thể dùng chớp, nháy).
- Không dùng quá nhiều màu sắc trên một màn hình.

## ■ Nhất quán trong toàn bộ hệ thống về:

- Cách trình bày: font chữ, màu sắc, vị trí các mục.
- Tên các nút điều khiển.
- Ý nghĩa biểu tượng.



## 8. 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.

# 8. Đánh giá UI

## ■ 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?

## 8. Đánh giá UI

### ■ Các kỹ 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.

## 9. Thiết kế giao diện cho ứng dụng

1. Kết quả đạt được
2. Phân loại màn hình
3. Kiến trúc màn hình
4. Các bước thực hiện
5. Cách thiết kế các loại màn hình
6. Một số nguyên tắc trong thiết kế giao diện
7. Công cụ hỗ trợ

# 9. Thiết kế giao diện cho ứng dụng

## 9.1. Kết quả đạt được:

- Nội dung và hình thức trình bày các màn hình giao tiếp của phần mềm.
- Hệ thống các thao tác mà người dùng thực hiện trên màn hình giao tiếp và xử lý tương ứng của phần mềm.

# 9. Thiết kế giao diện cho ứng dụng

## 9.1. Kết quả đạt được:

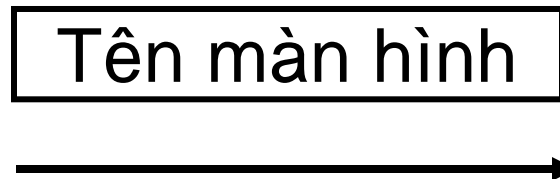
- Sơ đồ liên kết các màn hình
- Danh sách các màn hình
- Mô tả từng màn hình:
  - Mô tả các đối tượng trên màn hình;
  - Danh sách biến cố và xử lý tương ứng.

# 9. Thiết kế giao diện cho ứng dụng

## 9.1. Kết quả đạt được:

### 9.1.1 Sơ đồ liên kết các màn hình

- Hệ thống các màn hình cùng với quan hệ về việc chuyển điều khiển giữa chúng.
- **Hệ thống các màn hình** = Màn hình chính + Các màn hình thực hiện các công việc của phần mềm.
- **Ký hiệu:**



# 9. Thiết kế giao diện cho ứng dụng

## 9.1. Kết quả đạt được:

### 9.1.2 Danh sách các màn hình

TT	Màn hình	Loại màn hình	Chức năng
1	Màn hình Tiếp nhận bảo trì xe	Màn hình nhập liệu	Cho phép nhập và lưu trữ thông tin về các xe sửa chữa.
2	Màn hình Lập phiếu sửa chữa	Màn hình nhập liệu	Cho phép nhập và lưu trữ thông tin về các phiếu sửa chữa.
3	Màn hình Tra cứu xe	Màn hình tra cứu	Cho phép nhập <b>các tiêu chuẩn tra cứu</b> và trình bày kết quả tra cứu được.
4	Màn hình Lập phiếu thu tiền	Màn hình nhập liệu	Cho phép nhập và lưu trữ thông tin về các phiếu thu tiền.
5	Màn hình Báo cáo tháng	Báo biểu	Trình bày kết quả báo cáo.



# 9. Thiết kế giao diện cho ứng dụng

## 9.1. Kết quả đạt được:

### 9.1.3 Mô tả chi tiết từng màn hình

#### ■ Mô tả các đối tượng trên màn hình

STT	Tên	Kiểu	Ràng buộc	Chức năng
1	txtSĐT	TextBox	Nhập ký số 0-9	Nhập số điện thoại của chủ xe.

#### ■ Danh sách biến cố và xử lý tương ứng trên màn hình

STT	Biến cố	Xử lý
1	Chọn button Lưu	Lưu thông tin về xe sẽ sửa chữa xuống CSDL.

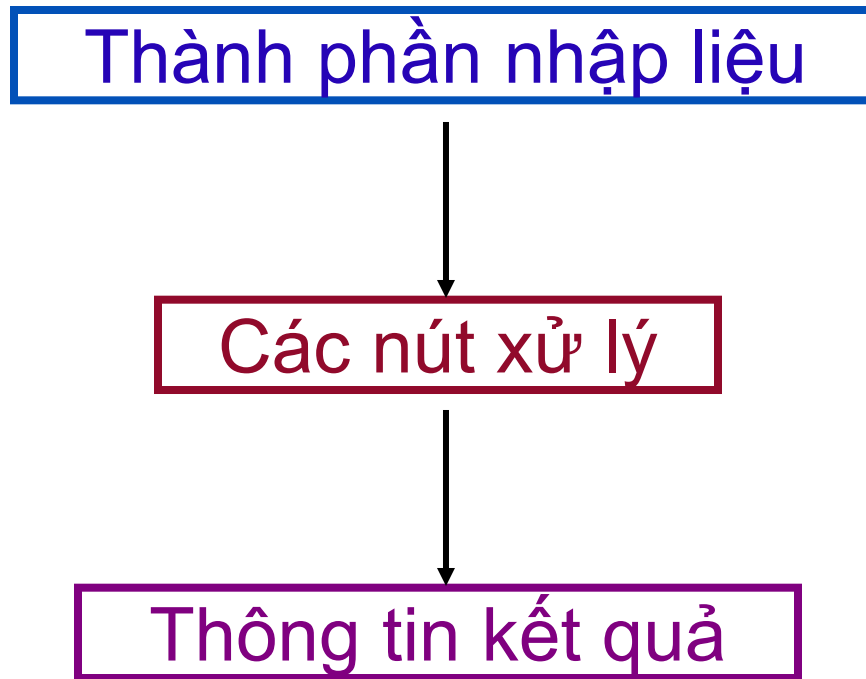
# 9. Thiết kế giao diện cho ứng dụng

## 9.2. Phân loại màn hình

- **Màn hình chính:** cho phép người sử dụng chọn các công việc sẽ thực hiện với phần mềm.
- **Màn hình nhập liệu:** cho phép người sử dụng nhập vào các thông tin để lưu trữ, tính toán.
- **Màn hình tra cứu:** cho phép tìm kiếm thông tin đã được lưu trữ với các tiêu chuẩn tìm kiếm.
- **Màn hình thông báo:** hiển thị các thông báo, nhắc nhở.
- **Báo biểu:** các báo cáo thống kê theo một mốc thời gian định sẵn.

# 9. Thiết kế giao diện cho ứng dụng

## 9.3. Kiến trúc màn hình



# 9. Thiết kế giao diện cho ứng dụng

## 9.3. Kiến trúc màn hình

### 9.3.1 Thành phần nhập liệu

- Cho phép người sử dụng nhập dữ liệu dưới nhiều hình thức khác nhau:
  - Text Box
  - Combo Box
  - List Box
  - Radio Button/Option Button
  - Check Box/Tick Box

# 9. Thiết kế giao diện cho ứng dụng

## 9.3. Kiến trúc màn hình

### 9.3.2 Các nút xử lý

- Các nút xử lý cho phép người sử dụng yêu cầu phần mềm thực hiện một xử lý nào đó.
- **Tên các nút xử lý:**
  - Không quá dài;
  - Gợi nhớ;
  - Nhất quán trong toàn hệ thống.

# 9. Thiết kế giao diện cho ứng dụng

## 9.3. Kiến trúc màn hình

### 9.3.3 Thông tin kết quả

- Cho phép người sử dụng xem thông tin kết quả dưới nhiều hình thức khác nhau:
  - Label
  - Text Box
  - List Box

# 9. Thiết kế giao diện cho ứng dụng

## 9.4. Các bước thực hiện

**Sơ đồ luồng dữ liệu/Sơ đồ lớp chi tiết, biểu đồ cộng tác**



**Thiết kế màn hình với tính đúng đắn**



**Màn hình đúng đắn**

**Thiết kế màn hình với tính tiện dụng**



**Màn hình hiệu quả      Màn hình tiện dụng**



**Thiết kế màn hình với tính hiệu quả**



**Màn hình kết quả**

# 9. Thiết kế giao diện cho ứng dụng

## 9.4. Các bước thực hiện

### 9.4.1 Thiết kế màn hình với tính Tiện dụng

- Giao diện quen thuộc (dựa trên biểu mẫu tương ứng).
- Bố trí hợp lý.
- Cung cấp thêm thông tin cho người dùng.
- Cho phép nhập nhiều giá trị đồng thời (trên List Box).



# 9. Thiết kế giao diện cho ứng dụng

## 9.4. Các bước thực hiện

### 9.4.2 Thiết kế màn hình với tính Hiệu quả

- Chọn control thích hợp (Text Box/Combo Box...).
- Cung cấp giá trị mặc định cho ô nhập liệu.
- Hiển thị kết quả một cách trực quan (dùng màu sắc, hình vẽ, chú thích...).

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

- Màn hình chính
- Màn hình nhập liệu
- Màn hình tra cứu
- Màn hình thông báo
- Báo biểu

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.1 Thiết kế màn hình chính

- Từ danh sách các yêu cầu phần mềm (nghiệp vụ, chất lượng và hệ thống), tiến hành phân nhóm các công việc:
  - Theo chức năng Tin học:
    - Hệ thống
    - Lưu trữ
    - Tra cứu
    - Báo biểu

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.1 Thiết kế màn hình chính

- **Theo đối tượng:** danh sách các đối tượng và các công việc sẽ thực hiện trên từng đối tượng.
- **Theo nghiệp vụ, qui trình:** trình bày các công việc theo trình tự của công việc trong qui trình.

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.1 Thiết kế màn hình chính

#### ■ Lựa chọn hình thức trình bày:

- Thực đơn (kết hợp sử dụng phím nóng)
- Biểu tượng
- Sơ đồ

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.2 Thiết kế màn hình Nhập liệu

- **Thành phần nhập liệu:** kiểm tra **tất cả dữ liệu nhập** dựa vào **ràng buộc tự nhiên** và **ràng buộc nghiệp vụ** (qui định liên quan).

***Thông thường mã số được phát sinh tự động.***

- **Các nút xử lý:**

- Thêm mới, Lưu...
- Bổ sung các nút chuyển điều khiển đến màn hình có liên quan.

- **Thông tin kết quả:** trình bày kết quả nhập liệu.

Bổ sung **thông tin tính toán** (số sách đã mượn, tổng số tiền hiện tại của hóa đơn...).

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.3 Thiết kế màn hình Tra cứu

- Cho phép tra cứu theo ***nhiều tiêu chuẩn khác nhau***, ưu tiên ***chọn các tiêu chuẩn tra cứu quen thuộc và dễ dàng*** cho người sử dụng.
- Cho phép ***tìm và cập nhật lại thông tin đã lưu trữ***.
- Kết quả tra cứu = Thông tin của đối tượng và ***quá trình hoạt động của đối tượng***.
- Hỗ trợ xem kết quả tra cứu dưới nhiều hình thức khác nhau (danh sách, hình ảnh...).
- Bổ sung các nút điều khiển cho nhu cầu kết xuất ra máy in, tập tin...

# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.4 Thiết kế màn hình thông báo

- Thông báo về **kết quả thực hiện yêu cầu** (thành công/***thất bại + nguyên nhân***).
- Cung cấp phản hồi của hệ thống khi cần thiết (progress bar, thông báo chờ khi hệ thống đang thực hiện một xử lý mất nhiều thời gian...).
- Thiết kế đơn giản, gồm thông tin cần thông báo (cô đọng nhưng dễ hiểu) và các nút chọn.
- Không thiết kế quá nhiều nút chọn.
- Chọn nút mặc định theo cách xử lý thông thường.



# 9. Thiết kế giao diện cho ứng dụng

## 9.5. Cách thiết kế các loại màn hình

### 9.5.5 Thiết kế Báo biểu

- **Giữ lại tiêu đề** báo cáo khi qua trang khác hoặc khi kéo thanh trượt lên xuống.
- Lưu ý kích thước khác nhau về không gian hiển thị giữa **báo biểu in ra giấy** và **báo biểu xuất ra màn hình**.
- Chỉ hiển thị những thông tin thật sự cần thiết, tránh làm rối báo biểu.

# Hội Đồng

Các Hội Đồng hiện có:

Mã: HD01

Tên: HD1

Danh sách đề tài được Hội đồng xét duyệt:

Mã đề tài	Tên Đề tài	Ngày Bắt Đầu...	Ngày kết thúc...	Kết quả
DT001	Nghiên cứu...	7/22/2004...	7/22/2004...	1
DT003	Phần mềm	7/22/2004...	7/22/2004...	1
DT004	Phần cứng	7/15/2004...	7/16/2004...	1
DT005	Ấm thực	7/14/2004...	7/15/2004...	1
▶ DT006	Lịch sử thời cổ...	7/15/2004...	7/16/2004...	1
DT007	Đặc nhân tâm	7/14/2004...	7/15/2004...	1
DT008	Quản lý thư...	7/22/2000...	7/22/2000...	1
DT009	a	7/22/2004...	7/22/2004...	1

Thông tin Hội Đồng:

Mã Hội Đồng: HD01

Tên HĐ: HD1

Số Chuyên Gia: 3

Thêm

Cập nhật

Xóa

Phân công xét duyệt đề tài

Thoát

Phân Công chuyên gia vào Hội Đồng:

DS Chuyên Gia không thuộc Hội Đồng

Mã	Họ Tên	Lĩnh Vực
▶ CG004	Yến Phương	Văn học
CG005	Vĩnh Phúc	Nghệ thuật
CG006	Tuấn Thành	Sử học
CG007	Linh Phương	Sinh học
CG008	An Nghi	Công nghệ
CG009	Huỳnh Nhân	Nghệ thuật
CG010	Sinh	Nghệ thuật
CG011	Thanh	Công nghệ

>>

<<

Tham gia 0 hội đồng

DS Chuyên Gia thuộc Hội Đồng:

Mã	Họ Tên	Lĩnh Vực
▶ CG001	Mạn Ngọc	Toán học
CG002	Phú Thành	Công nghệ
CG003	Nhật Hoa	Công nghệ

## Phân công Hội đồng xét duyệt Đề tài

Mã Hội đồng: HD04

Ngày bắt đầu  
xét duyệt: 8/ 1/2004

Tên Hội Đồng: HD4

Số chuyên gia: 3

Mã Đề Tài	Tên Đề Tài	Lĩnh Vực
DT007	Đột biến Tiến...	Công nghệ...
DT008	Cơ sở thực...	Công nghệ...
DT010	Quá trình tiến...	Khoa học vũ trụ
DT011	Tính bất toàn...	Khoa học vũ trụ
DT013	Tính liên tục...	Khoa học vũ trụ
DT015	Du lịch và di...	Văn Hóa

>>

<<

Mã đề tài	Tên Đề tài	Ngày Bắt Đầu...	Ngày kết thúc...
DT006	Khai thác...	8/1/2004...	
DT014	Kỹ thuật...	7/1/2001...	7/1/2001...
DT017	Thừa kế và di...	7/1/2001...	7/31/2004...

Nhập kết quả xét duyệt

Thoát



## Chuyên Gia

Mã Chuyên Gia:

CG009

Họ Tên:

Huỳnh Nhân

Phái.....

☐ Nam☒ Nữ

Quốc tịch.....

VIỆT NAM

Lĩnh Vực trực thuộc.....

Nghệ thuật

Địa Chỉ:

Nguyễn Văn Cừ

Danh sách các chuyên gia:


Mã	Họ Tên
CG001	Mạn Ngọc
CG002	Phú Thành
CG003	Nhật Hoa
CG004	Yến Phương
CG005	Vĩnh Phúc
CG006	Tuấn Thành
CG007	Linh Phương
CG008	An Nghi
CG009	Huỳnh Nhân
CG010	Đinh

Thêm

Cập nhật

Xóa

Thoát



**Q & A**

# Ôn tập

- Các nguyên lý Các vấn đề khi thiết kế UI giúp Các vấn đề khi 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.

# Ôn tập

- 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 nguyên tắc thiết kế UI
- 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.

# Ôn tập

- Thiết kế giao diện ứng dụng:
  - ◆ Hãy nêu các loại màn hình.
  - ◆ Kiến trúc của màn hình gồm có những thành phần nào?
  - ◆ Các bước thiết kế một màn hình.
  - ◆ Số bước để đi đến một màn hình công việc chính?