



# Thành phần thiết kế mobile

Hiển thị, Điều khiển, Thông tin

# Mở đầu

- Hiện thị thông tin
- Điều khiển và xác định
- Thể hiện về thông tin

# Hiển thị thông tin

- Các kiểu hiển thị thông tin
- Phân loại thông tin
- Các mẫu hiển thị
- Vertical List
- Infinite List
- Thumbnail List
- Grid
- Film Strip

# Các kiểu hiển thị thông tin

- Các mối quan hệ thông tin được xác định qua cấu trúc và các mẫu nằm trong quan hệ qua lại lẫn nhau.
- Quan hệ thông tin có thể là cấu trúc, khái niệm, nhân quả và thời gian.
- Các thực thể và quan hệ có thể được mô tả là các thuộc tính của thông tin. Một số thuộc tính là:
  - Màu sắc
  - Khoảng thời gian
  - Cấu trúc
  - Mức độ thể hiện thông tin theo dòng (độ dày, mỏng)
  - Kiểu kích thước

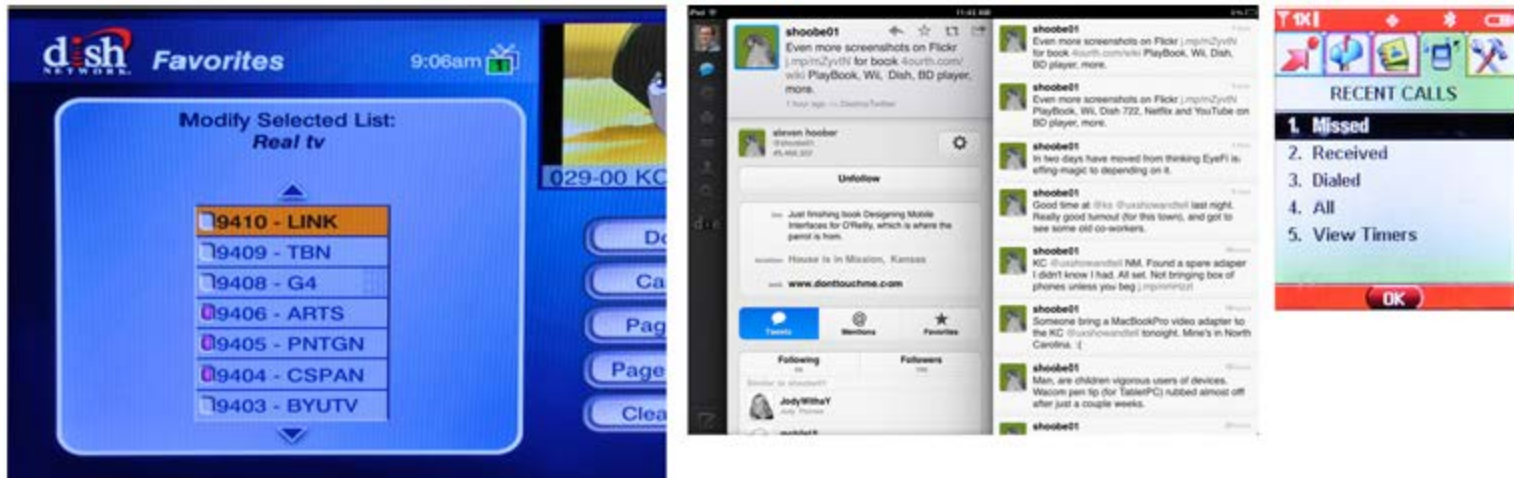
# Phân loại thông tin

- Thông tin được chia ra làm các loại sau:
  - Thông tin dạng định danh – là các nhãn và tên của các category dữ liệu.
  - Thông tin dạng thứ tự - là các số nhằm sắp xếp thứ tự theo một trình tự nào đó.
  - Thông tin dạng ratio (theo tỷ lệ) – là mối quan hệ cố định giữa một đối tượng đối với đối tượng khác trong sử dụng dữ liệu rộng như là một tham chiếu.
  - Thông tin dạng khoảng – là sự đo khoảng giữa 2 giá trị dữ liệu.
  - Thông tin dạng thứ tự chữ cái – sử dụng để sắp xếp theo alphabet nhằm mô tả thông tin định danh.

# Phân loại thông tin

- Thông tin dạng vị trí – xác định vị trí theo bản đồ như thành phố, tỉnh ...
- Thông tin dạng chuyên đề - tổ chức dữ liệu theo chuyên đề hoặc đối tượng.
- Thông tin dạng tác vụ - tổ chức dữ liệu dựa trên tiến trình, nhiệm vụ, chức năng và mục đích.
- Thông tin dạng thính giả - tổ chức dữ liệu theo kiểu người dùng, như thông qua thói quen, sở thích, sự hiểu biết và kinh nghiệm.
- Thông tin dạng xã hội – liên kết của các tổ chức dữ liệu theo người dùng, khi nhóm lại các đối tượng người dùng theo sở thích, kinh nghiệm ...
- Thông tin dạng ẩn dụ - tổ chức thông tin dựa trên mô hình mà đồng nhất tới người dùng, như tổ chức của các file máy tính với thư mục, thùng rác ...

# Tổ chức thông tin



- Kể cả trong phác họa hoặc thiết kế chi tiết giao diện, con người có thể liệt kê các thành phần tương tác trong thiết bị di động.
- Sự liệt kê có thể là vô hạn, là sự tạo mô hình, hoặc lựa chọn cho bất kỳ kích cỡ, và bất kỳ kiểu tương tác nào.

# Tổ chức thông tin

- Để hiểu được thông tin được thể hiện ra sao, cần phải biết về cấu trúc, tổ chức, nhãn, và các tính đặc trưng trong tương tác trên di động.
- Một trong số các vấn đề đó là phải thể hiện được sự phân cấp thông tin.
- Thông tin có thể được phân cấp theo category nhưng phải cân bằng giữa chiều rộng và độ sâu của thông tin.
- Chiều rộng của thông tin là mô tả được số lượng thông tin, đánh số thứ tự các thông tin theo kiểu, loại.
- Độ sâu của thông tin là mức độ thể hiện được các thuộc tính của thông tin, chức năng thông tin, sự nhập nhằng thông tin.



# Tổ chức thông tin

- Một cách khác để tổ chức thông tin là thể hiện theo khía cạnh của thông tin.
- Theo cách này, sẽ không có quan hệ cha-con của thông tin. Thông tin chỉ được xác định theo các thuộc tính, mà có thể được sắp xếp hoặc chọn lọc để hiển thị.
- Cách tổ chức này được thể hiện các dạng thông tin theo tìm kiếm, thông tin về các hình vẽ, mẫu, hoặc màu sắc.
- Hai kiểu tổ chức trên có thể được sử dụng đan xen nhau. Ví dụ như trong các thông tin về thời gian, địa điểm (có thể nhận giá trị bất kỳ, và cũng có thể phải được mô tả một cách chặt chẽ)

# Thiết kế thông tin và thứ tự dữ liệu



- Grid được sử dụng để thể hiện thứ tự hiển thị như hình ảnh theo ngày tháng.
- Sử dụng tổ chức thông tin như trang chủ.
- Grid còn được dùng trong thể hiện sự lựa chọn, tái tổ chức, có thể đưa thêm thông tin chừng nào còn vừa với grid.

# Thiết kế thông tin

- Thiết kế thông tin là thiết kế các trang hoặc các trạng thái nhằm thể hiện kiến trúc của tất cả các thông tin của toàn sản phẩm.
- Các hình thái của thiết kế thông tin bao gồm:
  - Vị trí – mỗi thông tin phải được thể hiện theo vị trí xác định, dựa trên layout của nó.
  - Kích thước – các phần tử lớn hơn sẽ thu hút được sự chú ý, và phải được cung cấp nhiều khoảng hiển thị hơn.
  - Hình dạng – là điểm thu hút sự chú ý. Như cảnh báo được thể hiện là hình tam giác, trợ giúp – hình icon tròn ...

# Thiết kế thông tin

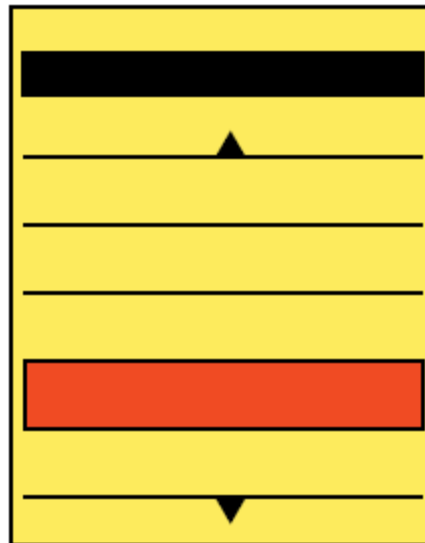
- Độ tương phản – là sự thu hút người dùng bằng các hiệu ứng ánh sáng, có thể tạo sự dễ đọc hoặc thu hút vào các mục đích hiển thị khác nhau.
- Màu sắc – thể hiện các đối tượng thông tin khác nhau.
- Dạng form – kiểu chữ, ký tự thể hiện thông tin như kiểu bold, italics.
- Trong thiết kế và thể hiện dữ liệu là sự kết hợp của nhiều kiểu thể hiện thông tin khác nhau, dựa trên các hình thức thể hiện thông tin và mục đích chương trình.

# Các mẫu hiển thị

- Thông tin được hiển thị theo các dạng list, bao gồm:
  - Vertical List
  - Infinite List
  - Thumbnail List
  - Grid
  - Film Strip
  - Vùng không giới hạn

# Vertical List

- Mục đích – thường được dùng để thể hiện thông tin dạng text.
- Vertical List thể hiện thông tin trên màn hình view, và cho phép người dùng di chuyển màn hình view tới các vùng thông tin khác.



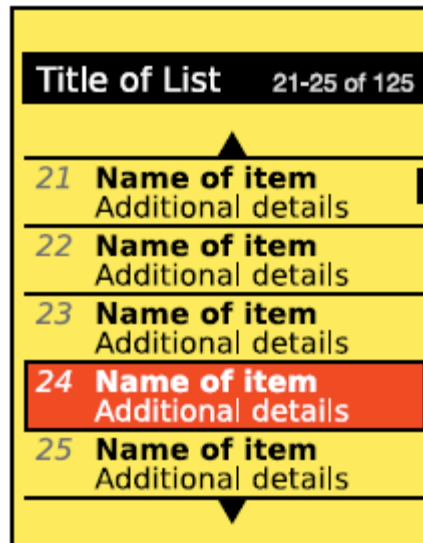
# Vertical List

- Vertical List kém hiệu quả trong việc cho phép sử dụng các khoảng view trên một dòng, do độ rộng của nó có thể chiếm toàn bộ chiều rộng của viewpoint.
- Các biến thể của vertical list có thể là:
  - Infinite list
  - Thumbnail list
  - Select list

# Vertical List

## ■ Tương tác:

- Tương tác trong vertical list phải mịn và mượt.
- Thông tin thể hiện qua tương tác phải rõ ràng.
- Từng item một trong list phải được thể hiện rõ.

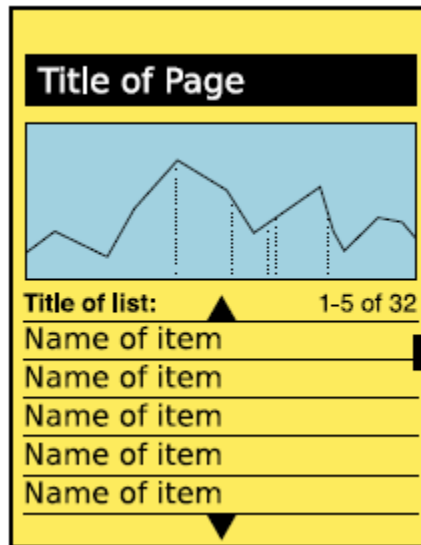




# Vertical List

- Sự lựa chọn phần tử trong list sẽ được mô tả theo hành động, như là xem chi tiết về phần tử được lựa chọn đó.
- Gia tốc scroll có thể được sử dụng cho một lượng lớn thông tin.
- Trong trường hợp là danh sách quay vòng, khi scroll tới phần tử kết thúc trong danh sách, nó sẽ tự động trở lại phần tử đầu.
- Trong trường hợp sử dụng dead-list sẽ không cho phép danh sách quay vòng, trừ khi cung cấp thêm tính năng Jump đến vị trí cho trước.

# Cách biểu diễn



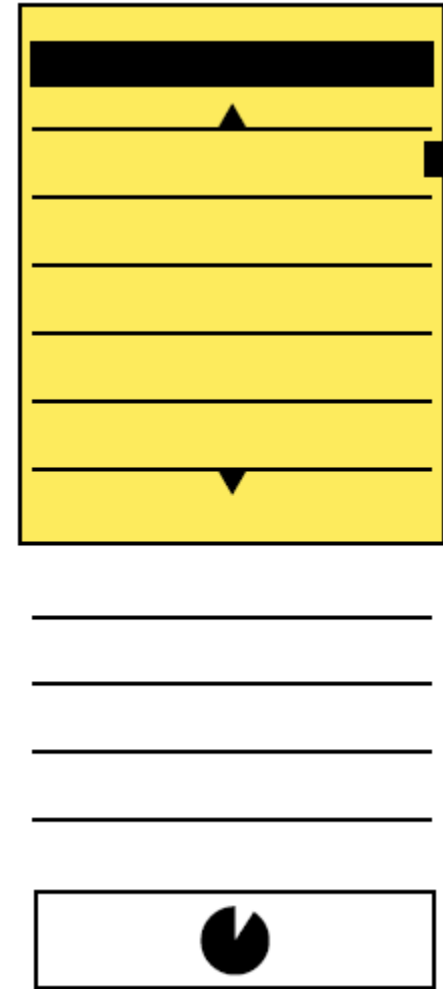
- Các thành phần của danh sách bao gồm:
  - Tiêu đề
  - Scrolling
  - Nhấn

# Các điểm chú ý

- Danh sách phải được cuộn theo các pixel của màn hình.
- Không được phép cuộn từng dòng nếu đó là giới hạn của thiết bị hoặc OS.
- Không bao giờ sử dụng cuộn trang.
- Khi tới phần tử cuối trong màn hình, phần tử tiếp theo trong dòng sẽ xuất hiện.
- Không được nhảy và load lại toàn bộ trang view của thông tin.

# Infinite List

- Là trường hợp thể hiện Vertical list đối với một lượng lớn các thông tin mà có thể cần phải load xuống từ các vùng lưu trữ khác.



# Infinite list

- Danh sách dạng này được sử dụng để nhận và hiển thị một lượng thông tin tại một điểm thời gian.
- Cần phải ưu tiên load thông tin ngoài vùng view khi có thể, nhằm giảm sự sự tương tác của người dùng trong việc load dữ liệu về.

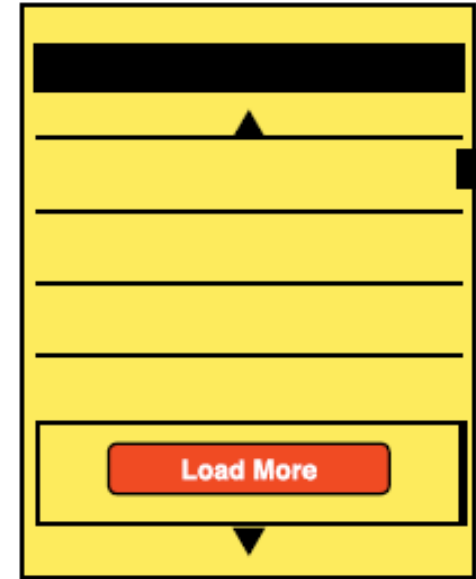
# Các biến thể



- Đoán trước về dữ liệu load về trước khi người sử dụng cần tới.
- Thường là sẽ có chỉ dẫn về loading dữ liệu ở phía dưới màn hình giao diện.
- Nhằm mục đích giảm thiểu băng thông của mạng.

# Các biến thể

- Có thể kết hợp infinite list với các dạng danh sách sau:
  - Thumbnail list
  - Selection list
  - Vertical list
- Khi gặp phần tử cuối của màn hình xác định của danh sách không giới hạn, thông tin phải được thể hiện cho người dùng biết là cần load thêm các dữ liệu.



# Các biến thể

- Một số thuộc tính cho infinite list:
  - Location jump – mục đích là có thể nhảy tới vị trí xác định trong danh sách.
  - Search within – tìm kiếm phần tử bên trong danh sách
  - Sort và bộ lọc – sắp xếp lại các thông tin bên trong danh sách, và tạo các bộ lọc cần thiết.
- Được áp dụng vào những giao diện:
  - Danh sách contacts,
  - Danh sách email,
  - RSS



# Cách biểu diễn

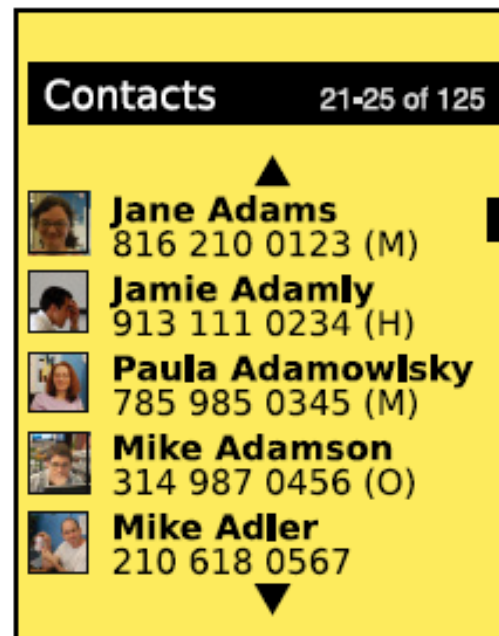
- Các hành động của infinite list là trong suốt với người sử dụng.
- Khi người dùng cuộn tới phần tử cuối cùng trong danh sách các thông tin đã nhận, cần phải thiết kế một vùng layout có thể nhìn thấy để cho người dùng load thêm các dữ liệu mới.
- Cần phải có thông báo lỗi trong quá trình load dữ liệu không được do đường truyền bị lỗi.
- Đối với dữ liệu không giới hạn, cần loại bỏ tất cả các chỉ dẫn như scroll bar, và sử dụng biến đếm cũng như gắn nhãn cho nó.

# Các điểm cần chú ý

- Cho tất cả các danh sách ưu tiên, phần tử cuối của danh sách có thể không nằm ở cuối. Nếu người dùng tới thông tin cuối khi cuộn lên, yêu cầu ưu tiên sẽ được đặt ở phần tử trên đầu danh sách.
- Hoàn toàn không có lý do thiết kế tốt nào cho yêu cầu của danh sách ưu tiên cho sắp xếp tùy ý.
- Chú ý tới việc chi phí của dữ liệu, khi mà load một lượng dữ liệu quá lớn cho danh sách.

# Thumbnail List

- Là trường hợp của Vertical list mà trong danh sách có thêm thông tin dạng hình ảnh.
- Thumbnail List sử dụng hình ảnh, icon, và thường được đặt ở bên trái của text, giúp làm rõ hơn sự khác biệt giữa các dòng.

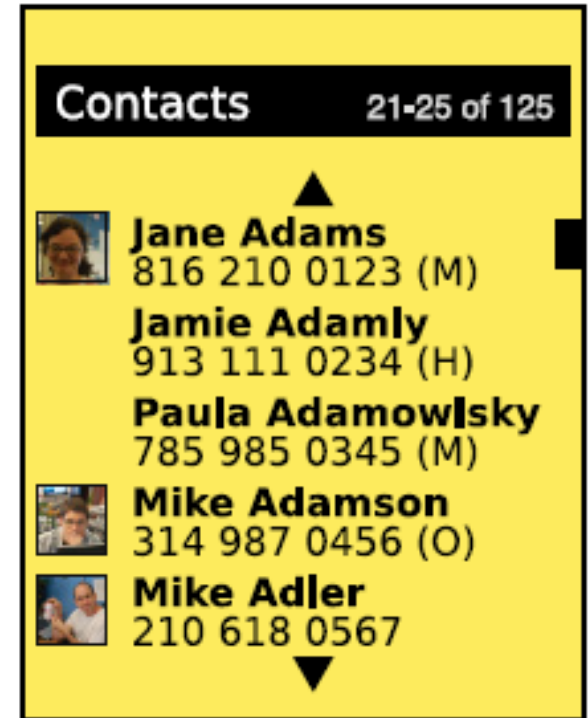


# Các biến thể

- Danh sách kiểu này được sử dụng để xác định và sắp xếp nếu có thể của các thành phần trong danh sách.
- Nếu bất kỳ một dòng nào không có hình ảnh, thì nó có thể load với sự bỏ trống ô hình ảnh của dòng đó.
- Có thể kết hợp danh sách kiểu này với các kiểu:
  - Select list
  - Infinite list

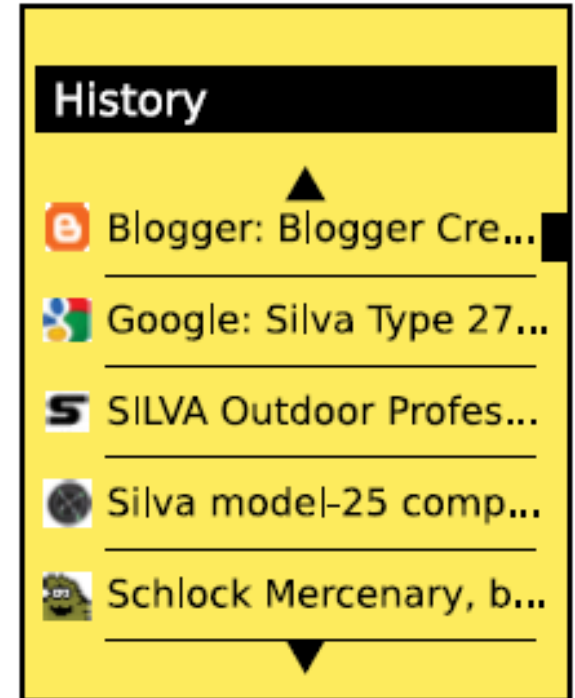
# Cách thể hiện

- Sự thay đổi thumbnail của bất kỳ dòng nào trong danh sách cũng không ảnh hưởng tới hình thức tương tác với danh sách đó.
- Cho phép người sử dụng tự định nghĩa các thumbnail.
- Nên xác định rõ kích thước hình ảnh khi cho vào thumbnail, vì nếu nhỏ quá nó sẽ tạo khoảng trống lớn với thành phần text



# Cách thể hiện

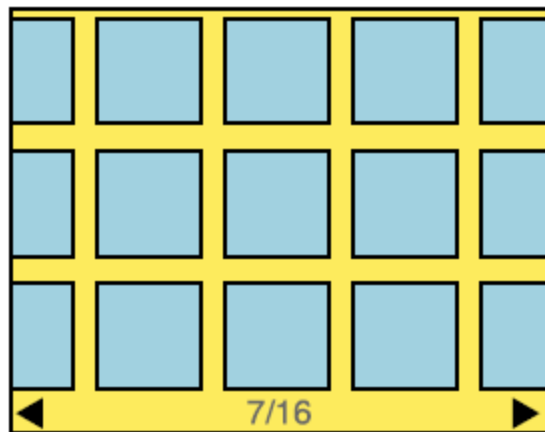
- Bất kỳ đối tượng hình ảnh nào cũng có thể được sử dụng làm thumbnail như các favicon ở các website.
- Không có định nghĩa hình ảnh cụ thể cho thumbnail, nhưng các hình ảnh phải được thể hiện đúng kích thước và nằm trong vùng bao cho phép.



# Các vấn đề cần lưu ý

- Tránh sử dụng mẫu này khi có ít những thumbnail. Ví dụ như address book, thumbnail có thể được áp dụng nhưng phần lớn là sử dụng với icon mặc định.
- Khi có rất ít hình ảnh mặc định, thì cần phải cân nhắc khi sử dụng chúng trong danh sách.
- Không được cắt hình ảnh nếu hình ảnh đó là sự nhận diện chính.
- Chú ý tới kích thước, màu sắc, khoảng cách và độ rộng của các thành phần trong danh sách.

# Grid



- Được sử dụng để thể hiện một tập hợp thông tin, mà phần lớn hoặc toàn bộ thành phần trong danh sách bao gồm các hình ảnh.
- Không yêu cầu phải thể hiện theo thứ tự.
- Có thể dễ dàng được sử dụng trong chương trình hoặc website trong bất kỳ thiết bị nào.

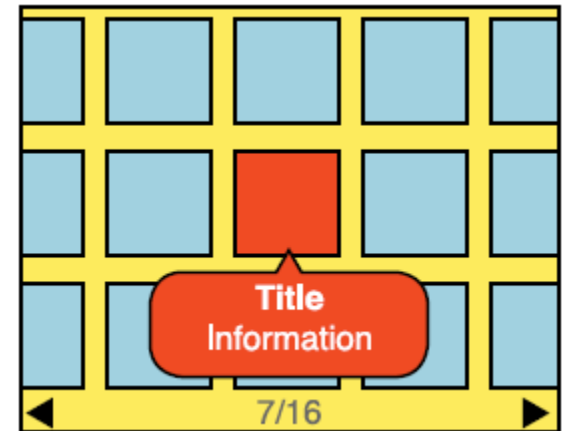


# Giải pháp

- Grid là tập hợp các hình ảnh lựa chọn được.
- Hình ảnh được thể hiện có thể giống nhau về kích thước và tỉ lệ.
- Người dùng có thể cuộn theo một số phương pháp.
- Chỉ báo có thể được thể phân biệt giữa các màn hình khác nhau.

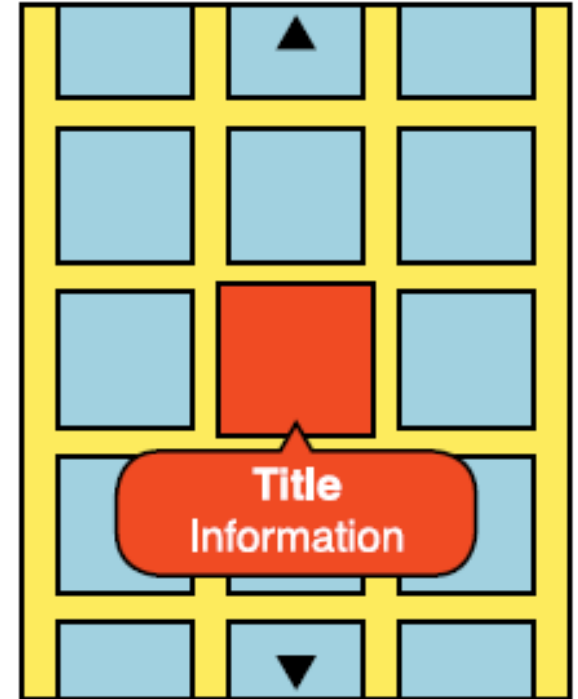
# Các biến thể

- Phần tử mà được lựa chọn phải rõ ràng, và phải có text đi kèm mô tả.
- Nên sử dụng Grid khi một trục tọa độ chuyển động theo một chiều view giữa các màn hình, và cấu trúc dữ liệu thể hiện.
- Có thể sử dụng một số effect trong grid như 2D, 3D.



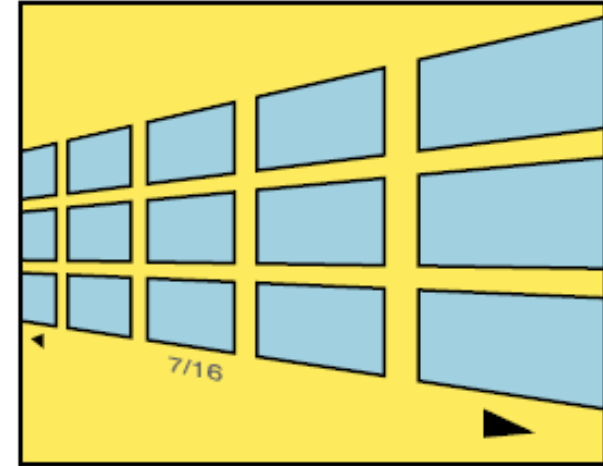
# Cách thể hiện

- Bất kỳ đối tượng được nhìn thấy trong grid đều có thể được lựa chọn.
- Có thể sử dụng một số hiệu ứng trong Grid:
  - Pop-up hoặc Slide show
- Có thể sử dụng hình thứ nhảy tới một vị trí nếu thông tin được mô tả tốt (có nghĩa là có sự sắp xếp, ghi nhớ ...)



# Cách thể hiện

- Các đối tượng có thể được chỉ báo và tập trung.
- Với các màn hình cảm ứng, thì Grid có thể được tương tác trực tiếp và chính xác.
- Một số hiệu ứng có thể áp dụng với cách thể hiện 3D.
- Hình ảnh ở cuối của màn hình view ngụ ý là sẽ có thêm những hình ảnh ở phần màn hình khác.
- Cuộn được sử dụng để di chuyển giữa các đối tượng trong grid.

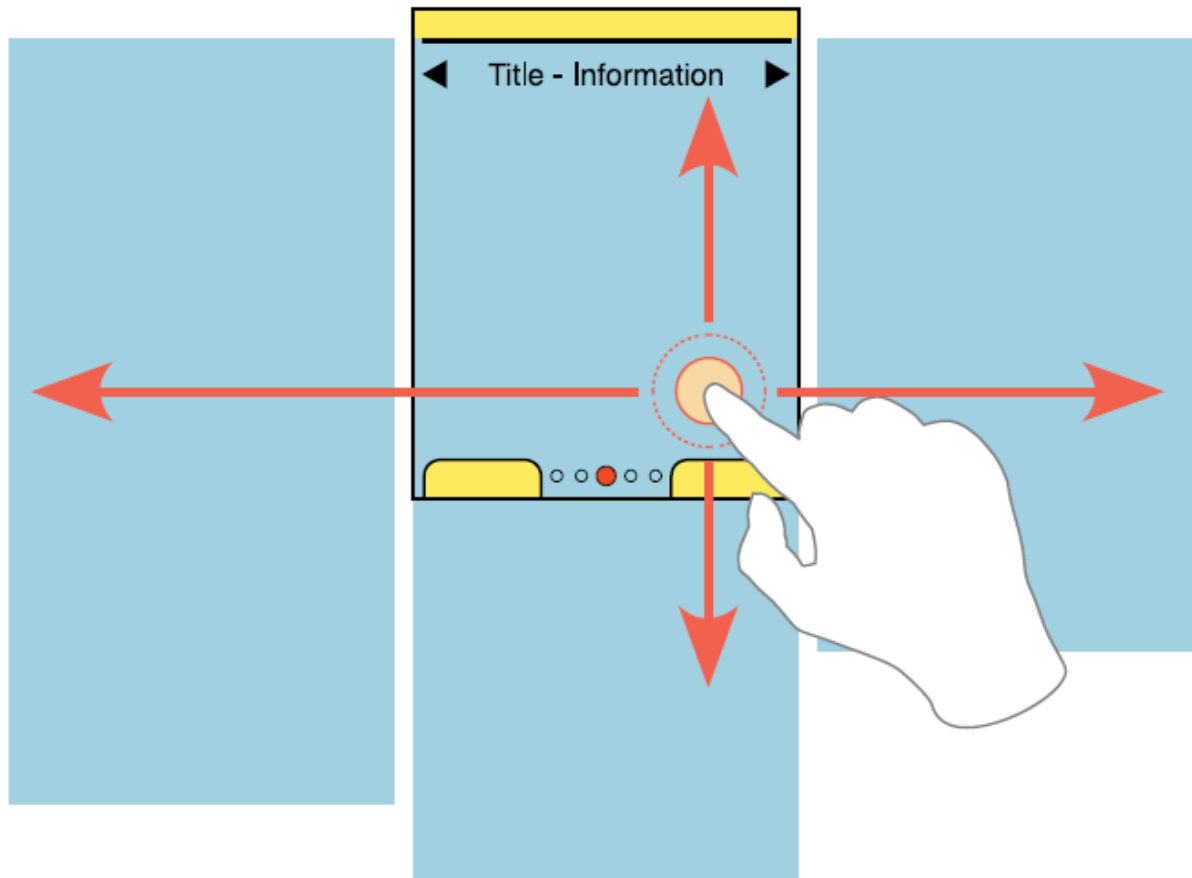


# Các vấn đề cần chú ý

- Không thể hiện các chức năng quá khó để tương tác với đối tượng trong grid.
- Tránh sự phức tạp trong hiển thị.
- Sử dụng grid khi màn hình và độ phân giải của thiết bị cho phép.
- Chắc chắn là cung cấp đủ khoảng không gian cần thiết giữa các đối tượng trong Grid.

# Film Strip

- Sử dụng thẻ hiển thị nhưng thông tin có trong một nhóm các màn hình view.



# Giải pháp

- Thể hiện các loại thông tin cần có sự liên tục, hoặc cần phải được thể hiện trong không gian lớn.
- Đối với thiết bị di động, mẫu này đơn giản là một chuỗi các màn hình thể hiện sự liên tục của thông tin (dải thông tin).

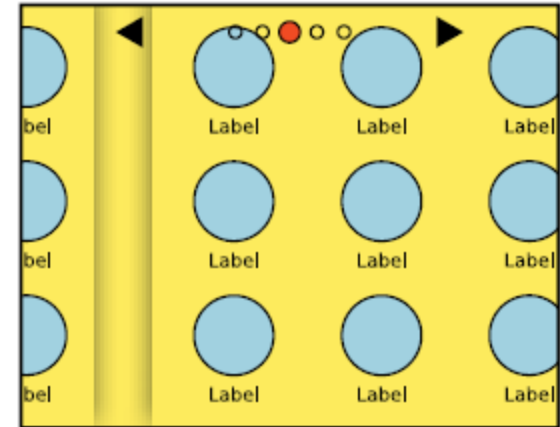
# Các biến thể

- Được thể hiện theo 2 cách:
  - Biểu diễn (hiển thị) các thành phần dạng full-frame, như hình ảnh trong mối quan hệ và được xác định rõ ràng.
  - Khi biểu diễn các đối tượng phức tạp, như kết hợp với grid hoặc nội dung của trang biểu diễn.
- Trong trường hợp cá biệt, mẫu này có thể là grid lớn, thông qua cuộn giữa các frames theo cả 2 hướng.

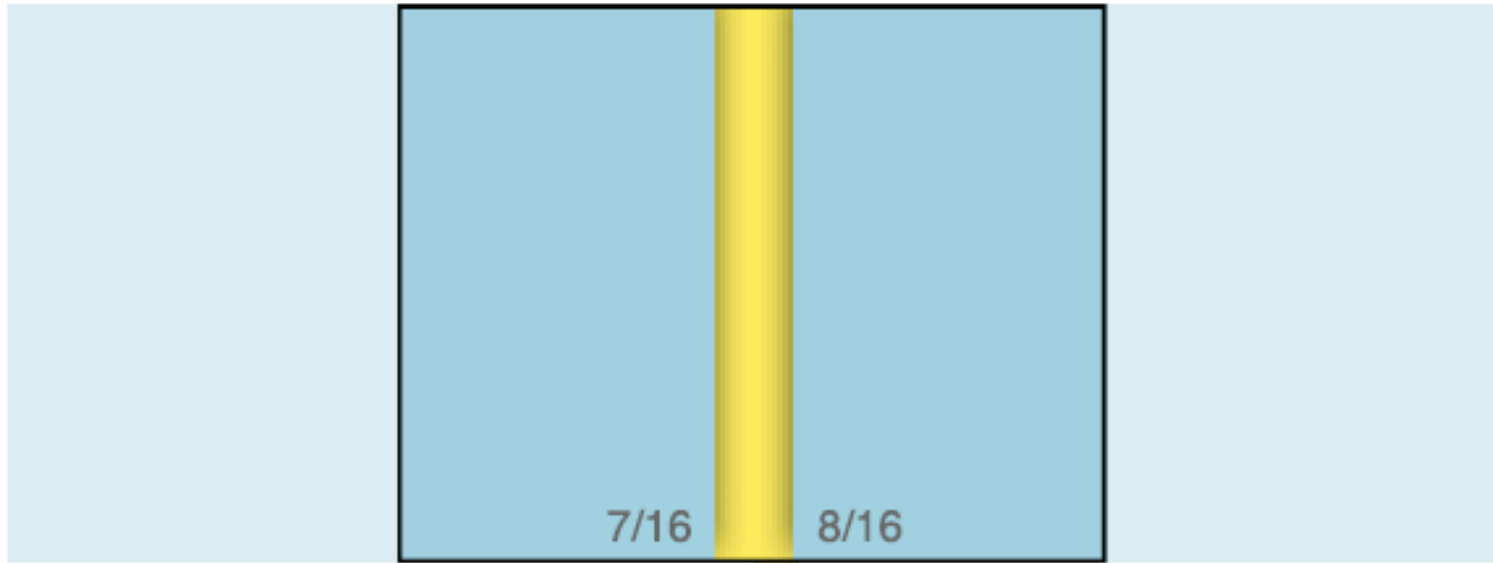


# Hình thức tương tác

- Có khoảng cách khi chuyển từ frame này sang frame khác.
- Có thể có một số hiệu ứng như bóng mờ xuất hiện khi slide các frame.
- Nếu sử dụng mẫu này đối với sự hiển thị phức tạp, không cần thiết phải cố định kích thước nó trong một viewpoint.



# Các cách thể hiện



- Phải miêu tả rõ khoảng giữa các frame riêng biệt.
- Mẫu phải được thể hiện phụ thuộc vào các kiểu dữ liệu, nó có thể được nhìn thấy trong quá trình cuộn, hoặc được nhìn thấy tại mọi thời điểm.

# Các điểm cần chú ý

- Không cho phép cuộn thẳng hay ngang trong một bảng đơn, ví dụ trong phóng to hình ảnh, cần phải vô hiệu hóa cuộn giữa các frames.
- Khi cuộn tới trang cuối, và chuyển động sang phải sẽ load lại trang đầu.

# Điều khiển và xác nhận

- Hiểu người sử dụng
- Các mẫu điều khiển
- Xác nhận

# Hiểu người sử dụng



- Người sử dụng có thể bị nhầm lẫn, mắc lỗi trong quá trình sử dụng.
- Giới hạn của người sử dụng trong việc mở rộng khả năng tương tác tới môi trường, và khả năng điều khiển tiến trình.
- Người sử dụng có xu hướng đưa những kinh nghiệm của mình vào trong quá trình sử dụng.

# Các mẫu điều khiển

## ■ Điều khiển:

- Liên quan tới sự tôn trọng dữ liệu người dùng và nhập liệu với mục đích đảm bảo tránh được các lỗi sử dụng thông thường, mất mát dữ liệu, và những quyết định không cần thiết.
- Đây là nguyên lý chính trong thiết kế mobile.

## ■ Xác nhận:

- Khi quyết định được đưa ra, cần phải có hành động lựa chọn các phương thức phòng tránh lỗi. Cần phải quan tâm tới những vấn đề:
  - Quyết định đó có cần thiết phải được xác nhận từ phía người sử dụng hay không?
  - Sẽ có sự loại bỏ những rủi ro của lỗi từ phía người dùng và mất mát dữ liệu input được thể hiện ra sao?

# Mẫu điều khiển và xác nhận

## ■ Xác nhận:

- Khi có quyết định xảy ra trong tiến trình mà người dùng phải xác nhận hành động, hoặc lựa chọn giữa số lượng nhỏ các lựa chọn.

## ■ Đăng nhập:

- Mẫu này được sử dụng để xác nhận chỉ duy nhất một quyền được cho phép truy cập vào thiết bị, site, dịch vụ hoặc các chương trình trên thiết bị.

## ■ Loại bỏ tính bảo mật:

- Mẫu này được sử dụng khi truy cập vào dữ liệu hoặc các tiến trình con nhằm giảm thiểu thời gian, hoặc sự khó chịu nhưng dễ dẫn tới mất mát dữ liệu.

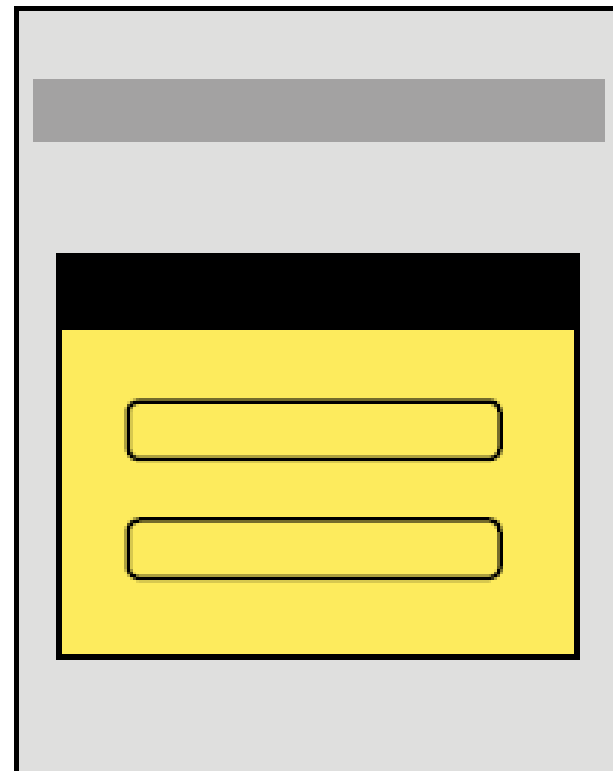
■ Timeout:

- Hệ thống bảo mật cao hoặc những hệ thống mà cho truy cập rộng và hệ thống chia sẻ nặng phải có thời gian để thoát khỏi session đó, hoặc khóa hệ thống sau một khoảng thời gian không hoạt động.



# Xác nhận

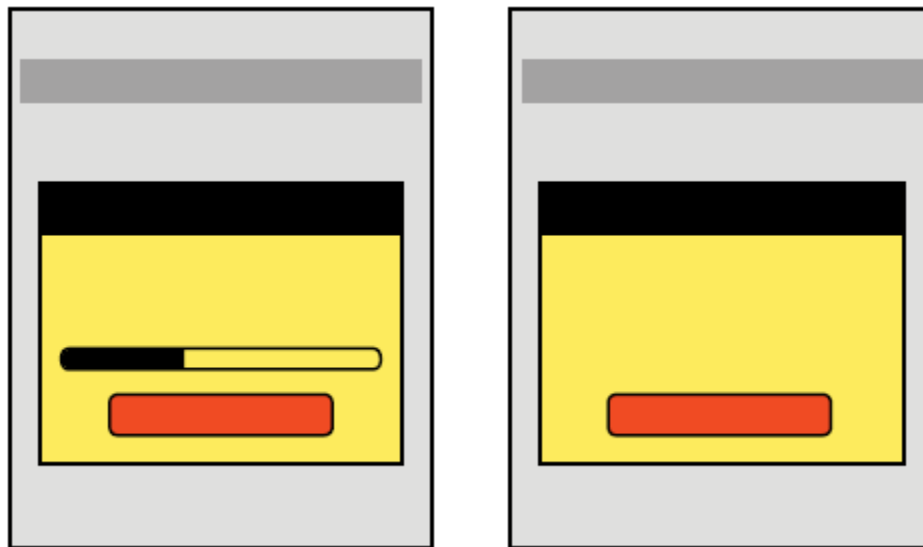
- Khi phải tạo tiến trình mà trong đó có một số điểm cần sự xác nhận của người dùng về hành động hoặc lựa chọn.
- Xác nhận thông qua một số bước đơn giản, và có thể được thực thi thông qua một số phương pháp.



# Giải pháp

- Có thể sử dụng các thông tin liên quan tới hành vi người dùng từ trước cho tới hiện tại, hoặc thông qua các nguồn khác để thể hiện lại những tùy chọn đúng đắn tới người dùng.
- Trong trường hợp phải cho người dùng lựa chọn giữa các chức năng trong luồng hoạt động của chương trình, phải làm cho lựa chọn đơn giản, và dễ hiểu.

# Biến thể

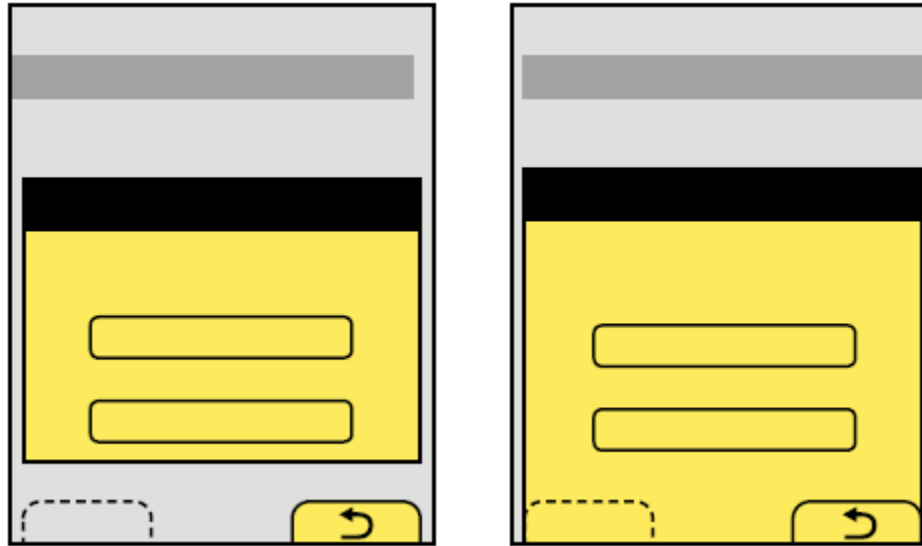


- Xác nhận lựa chọn đơn là các biến thể quan trọng.
- Người sử dụng phải được thông báo một điều gì đó mà không thể thay đổi, nhưng ảnh hưởng tới mình, như khi bị rớt cuộc gọi hoặc hệ thống bị treo.

# Biến thể

- Đưa ra dialog nhằm thông báo cho người dùng phải chờ đợi, hoặc xuất hiện nút hủy để thoát khỏi hệ thống (khi chỉ có duy nhất 1 sự lựa chọn).
- Các thông báo xác nhận cần phải có, kể cả khi người dùng đã loại bỏ các thông báo cơ bản khác từ thiết bị/OS.
- Ví dụ, khi soạn thảo và thoát khỏi chương trình mà không ghi lại, sẽ có thông báo lựa chọn:
  - Giữ lại chương trình để làm việc tiếp
  - Ghi lại văn bản và thoát
  - Thoát mà không ghi lại

# Tương tác



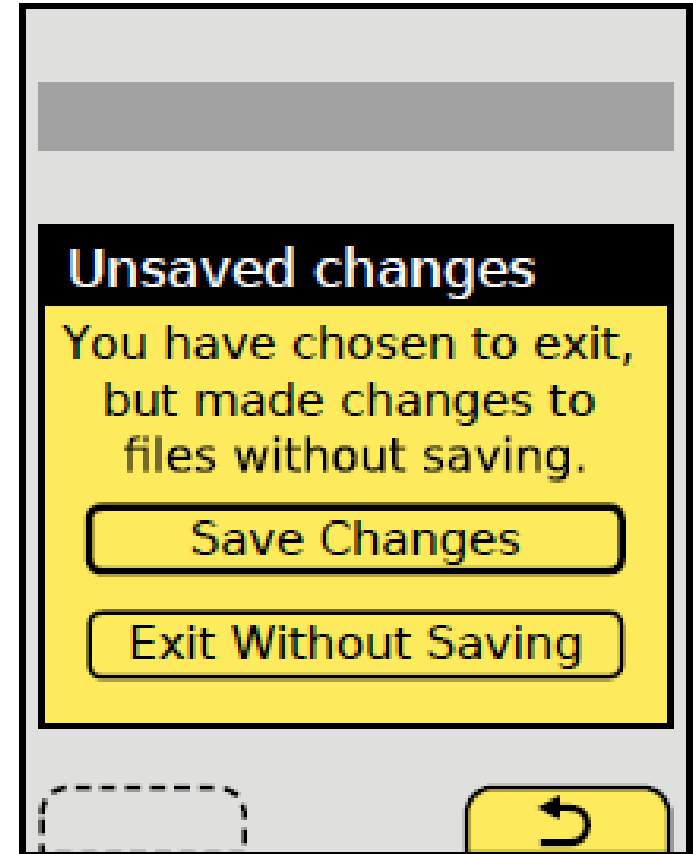
- Đối với thiết bị có phím mềm, có 2 phương pháp để thể hiện cửa sổ pop-up như trên hình vẽ.
- Cửa sổ pop-up có thể là động

# Tương tác

- Mẫu xác nhận liên quan tới lỗi hoặc lựa chọn để dừng tiến trình.
- Tốt nhất khi thiết kế, nên có từ 2 tới 3 sự lựa chọn đối với các tiến trình cần sự xác nhận của người sử dụng.
- Xác định sử dụng phím cứng hay mềm trong tương tác xác nhận.

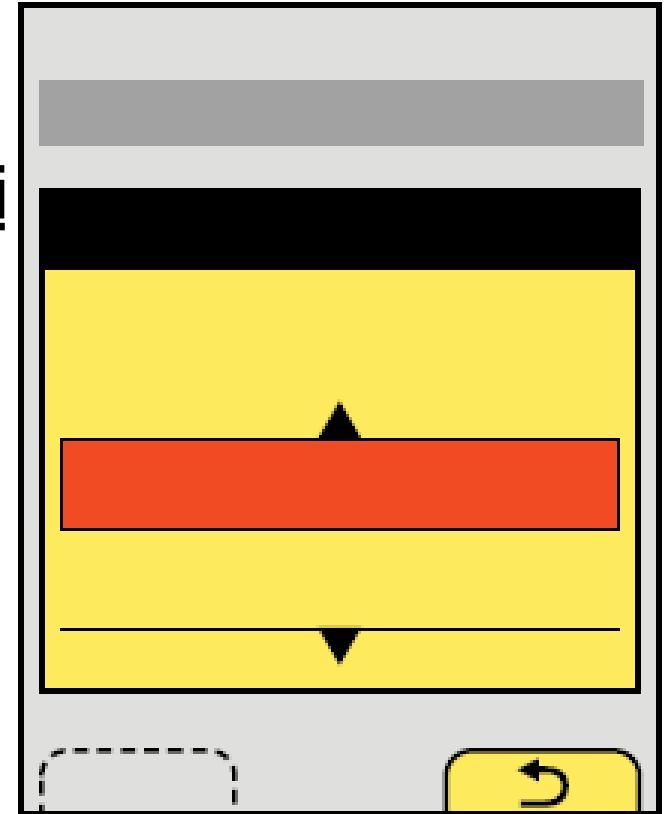
# Cách thể hiện

- Thông qua cửa sổ pop-up.
- Khi xác nhận liên quan tới các tiến trình background, cần phải sử dụng các thông báo mặc định của OS.
- Tiêu đề của dialog phải rõ ràng. Không được sử dụng nhãn tên là “Error” chung chung mà không đặc tả được thông báo.



# Cách thể hiện

- Trong trường hợp cần thông báo rõ ràng, mà thông tin cần thông báo vượt quá sự hiển thị của màn hình view của dialog, thì có thể sử dụng tính năng scroll.
- Cần phải viết ngắn gọn, dễ hiểu, không gây sự hoang mang cho người dùng trong lựa chọn sự xác nhận.





# Chi tiết cần lưu ý

- Giảm các lần click.
- Không được lạm dụng một lựa chọn trong thông báo.
- Cần thận khi cân nhắc thiết kế dialog xác nhận là đúng với mẫu chưa.
- Các thông báo xác nhận nên ít hơn 3 sự lựa chọn.
- Không được đưa nút “Close” vào dialog pop-up.
- Không được sáng tạo về ngôn ngữ cảnh báo, hoặc làm ngoài chuẩn.

# Sinh viên tự đọc sách

- Sinh viên đọc thêm các phần về Sign On, Exit Gurad, Cancel Protection và Timeout trong chương 3 quyển sách “Designing mobile interfaces”. Viết tóm tắt nội dung.
- Sinh viên đọc thêm chương 4 trong quyển sách “Designing mobile interfaces” và tóm tắt nội dung đọc được.

# Tổng kết

- Làm rõ hơn tính sử dụng của các thành phần thể hiện thông tin thông qua các mẫu thiết kế danh sách.
- Phương pháp tổ chức thông tin, lựa chọn hiển thị thông tin cho người dùng.
- Đánh giá dựa trên kinh nghiệm người dùng, và đưa ra một số chức năng điều khiển và xác nhận khi cần thiết đối với tiến trình trong chương trình.