



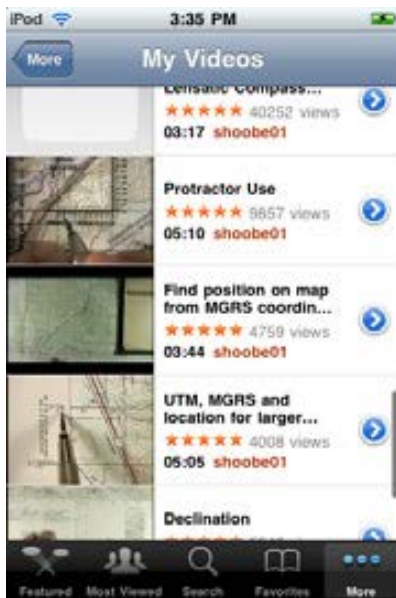
Thiết kế giao diện mobile

Mở đầu

- Nguyên tắc hợp thành (các thành phần) trên giao diện mobile
- Khái niệm về vùng bao
- Các thành phần mẫu
- Scroll
- Dòng thông báo
- Notifications
- Tiêu đề
- Các dạng menu
- Một số vấn đề khác về màn hình

Nguyên tắc hợp thành

- Các giao diện dưới đây chứa nhiều các thành phần khác nhau



Nguyên tắc hợp thành

- Hợp thành (composition) là một trong các tiến trình được hợp nhất trong giao diện cùng với các layouts và nội dung.
- Nguyên tắc hợp thành trở thành chuẩn mực trong thiết kế giao diện.
 - Các ràng buộc trong giao diện
 - Kiểu giao diện
 - Kích cỡ trang giao diện
 - Tỷ lệ các thành phần trong giao diện
- Các vấn đề trên sẽ được xem xét trong các mục tới

Khái niệm về vùng bao

- Người thiết kế tổ chức thông tin trong nhất quán theo hệ điều hành di động
- Thông tin được tổ chức theo sự phân cấp theo các trang giao diện
- Người sử dụng xác định cấu trúc tổ chức, học hỏi ghi nhớ về các thông tin theo trang, và quan tâm tới hiệu xuất làm việc, và các lỗi phát sinh

Vùng bao (wrapper)

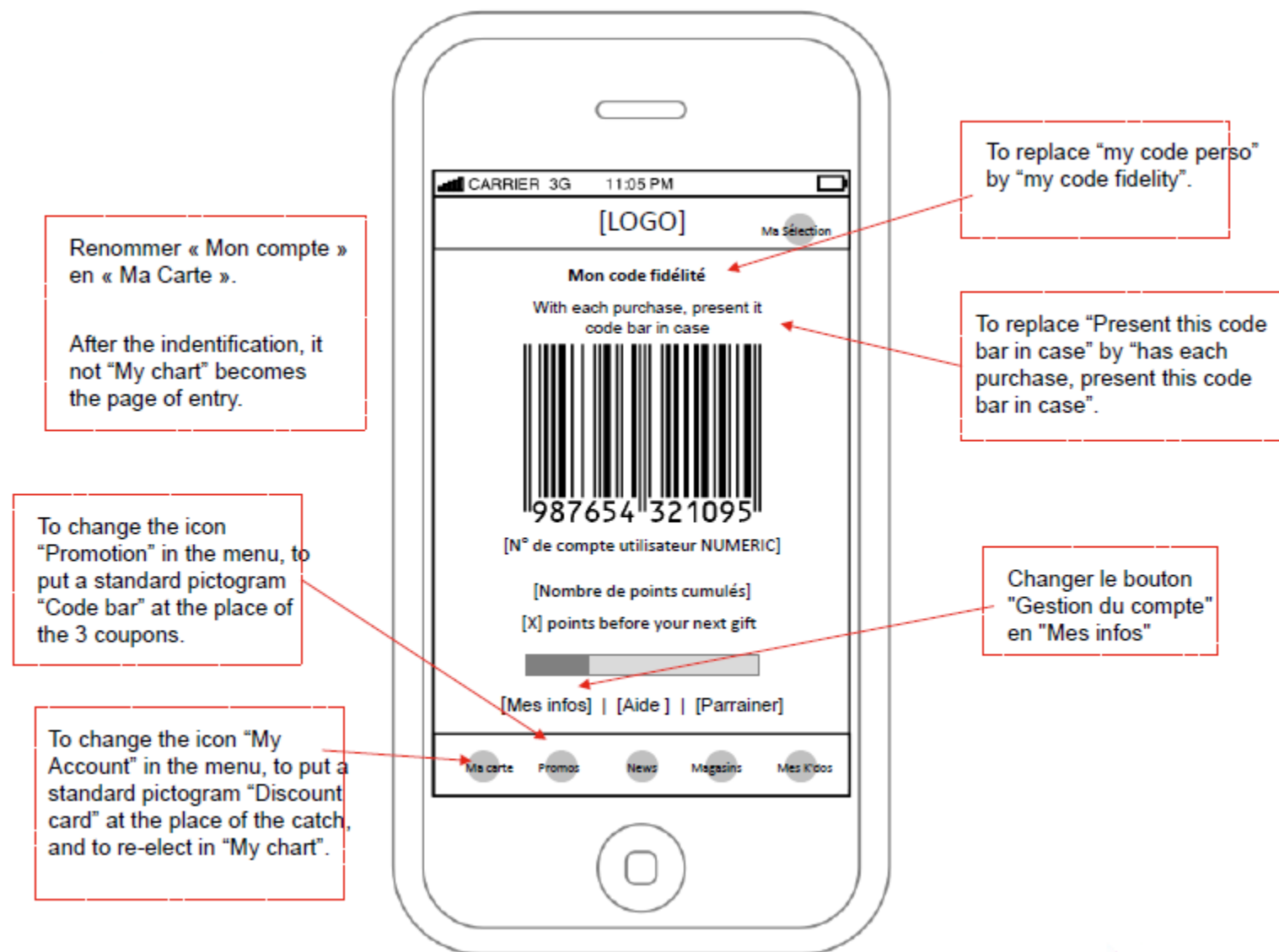


- Ví dụ như màn hình lock, thông tin được thể hiện qua tương tác cử động
- Vùng bao thể hiện qua grid, là thành phần quan trọng trong thiết kế, nhưng là duy nhất đối với từng chương trình

Vùng bao

- Wrapper phải được thiết kế dựa trên nội dung và trong ngữ cảnh sử dụng của các thành phần trong sản phẩm phần mềm
- Khi xác định các thông tin thuộc về wrapper, cần phải xác định rõ các phần sau:
 - Công nghệ, chức năng, và các yêu cầu, ràng buộc
 - Ngữ cảnh sử dụng (thông qua các kịch bản sử dụng)
 - Mục đích sử dụng của người dùng
 - Chức năng nào cần thiết để đạt được mục đích
 - Kiểu thông tin nào phải được thể hiện để đạt được mục đích và chức năng

Ví dụ



Các thành phần mẫu

- Sự sử dụng đúng và phù hợp của các vùng bao sẽ dẫn tới sự hiệu quả trong sử dụng và đánh giá kinh nghiệm người dùng
- Các thành phần mẫu trong thiết kế bao gồm:
 - Scroll – Khi thông tin được thể hiện vượt quá giới hạn view của trang, scroll bar sẽ hữu ích để có thể xem được các thông tin tiếp theo. Scroll luôn thể hiện theo một trục (trừ một số trường hợp đặc biệt).
 - Dòng thông báo – Thông báo trạng thái phần cứng tại phần đầu của mỗi trang view. Trạng thái phải được thể hiện thông qua radios, các thành phần vào ra, các mức sử dụng năng lượng.

Các thành phần mẫu

- Titles – Thể hiện tên gọi của từng trang view, nội dung, các thành phần cần có nhãn. Tên gọi được thể hiện theo bề ngang, phù hợp theo kiểu, hướng dẫn và khả năng đọc được.
- Menu theo quan hệ – Kiểu của menu phải được thể hiện dưới dạng ẩn. Nó có thể là cử chỉ, các phím ảo, hoặc được lựa chọn trên màn hình.
- Menu cố định – Kiểu menu này luôn phải được nhìn thấy hoặc điều khiển bởi các khung nhìn. Menu này phải được thể hiện ở vị trí phù hợp với chương trình. Tương tác thường thông qua các icon trên giao diện.
- Màn hình chính và màn hình nghỉ - Là các dạng màn hình thể hiện trạng thái khi thiết bị được bật hoặc tắt, hoặc là chương trình không sử dụng trong một khoảng thời gian.

Các thành phần mẫu

- Màn hình khóa – Là màn hình thể hiện trạng thái tiết kiệm năng lượng. Khi cần, màn hình này được kích hoạt để bảo vệ dữ liệu mà người dùng nhập vào.
- Màn hình xen kẽ - Kiểu màn hình này được sử dụng như để nạp các tiến trình trong quá trình khởi động thiết bị hoặc kích hoạt chương trình.
- Quảng cáo – Được sử dụng trong chương trình một cách riêng biệt, và không được ảnh hưởng tới chức năng người dùng. Quảng cáo phải được tuân theo hướng dẫn chuẩn của Mobile Marketing Association (MMA) (<http://www.mmaglobal.com/>)

Scroll

- Mục đích – để người dùng có thể xem được các thông tin trong khung nhìn trên một trang giao diện.
- Thông thường OS cung cấp chức năng này.
- Scroll có thể là một thanh bar hoàn thiện, hoặc là một điểm động đơn giản chạy theo nội dung thể hiện

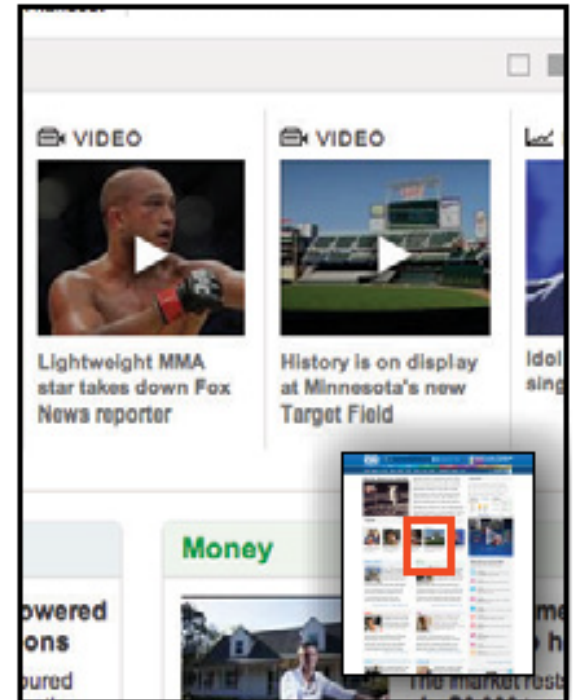


Scroll

- Scroll được sử dụng để thể hiện thông tin của các loại. Nó được thể hiện theo các trục, và liên quan chặt chẽ tới vị trí thông tin trong vùng thể hiện của nó.
- Scroll được sử dụng trong thể hiện thông tin dạng list hoặc list-like của các mẫu thông tin như:
 - Vertical list
 - Infinite list
 - Thumbnail list
 - Fisheye list
 - Carouel
 - Grid
 - Film Strip

Ví dụ

- Thể hiện Thumbnail list khi tập trung vào thể hiện một vùng trong một giao diện lớn.
- Được sử dụng để view các điểm và vị trí cố định trên giao diện.



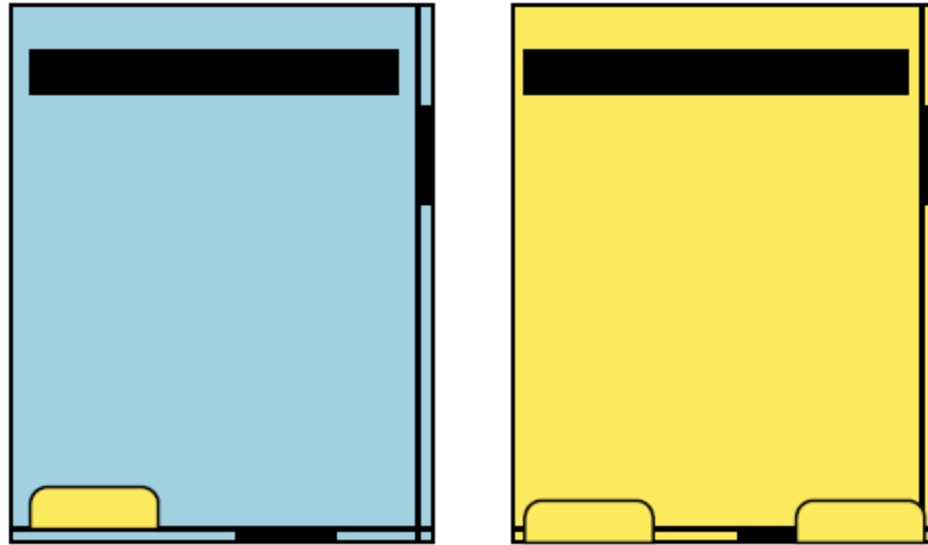
Scroll

- Trong một số trường hợp đặc biệt, scroll được sử dụng đối với cả 2 trục tọa độ, như trong các trường hợp view các ảnh phóng to.
- Các thành phần scroll có thể được sử dụng kết hợp với nhau
 - Như mặc định khi thiết kế là sử dụng Vertical scroll. Và có thể kết hợp với Horizontal scroll đối với các màn hình phụ.
- Thiết kế phải phù hợp với thiết bị sử dụng tương tác trực tiếp hoặc thông qua bút tương tác.
- Kích thước thường từ 5 – 10mm

Các hình thức scroll

- Scroll theo item – theo từng dòng trên giao diện
- Scroll theo chuyển động của các điểm trên màn hình.
- Scroll khi có sử dụng thêm các thiết bị pointer (như chuột, bút ...)
- Scroll theo kiểu link to link – thường được ứng dụng trong view các website (được sử dụng đối với thiết bị không có màn hình cảm ứng)
- Khi scroll, cần đảm bảo là các thông tin phải được cung cấp đầy đủ cho người dùng

Scroll



- Hai hình thức scroll theo 2 trục tọa độ
 - Ví dụ thể hiện thông tin dạng ảnh (hình bên trái)
 - Khi thông tin hướng theo 1 trục, và trục còn lại được sử dụng như là scroll phụ (và được làm mờ đi) như hình bên phải

Chỉ báo scroll

- Chỉ báo scroll (indicator) thường không được sử dụng nhiều, nhưng hữu ích trong:
 - Cung cấp thông tin về vùng mà được di chuyển tới
 - Truyền tải vị trí hiện tại trong tất cả các vùng thông tin
 - Chỉ báo về lượng thông tin được trong khung nhìn
- Chỉ báo có thể được thiết kế để nhìn thấy, hoặc là ẩn đi khi không có hành động về dịch chuyển.

Một số điểm tránh khi scroll

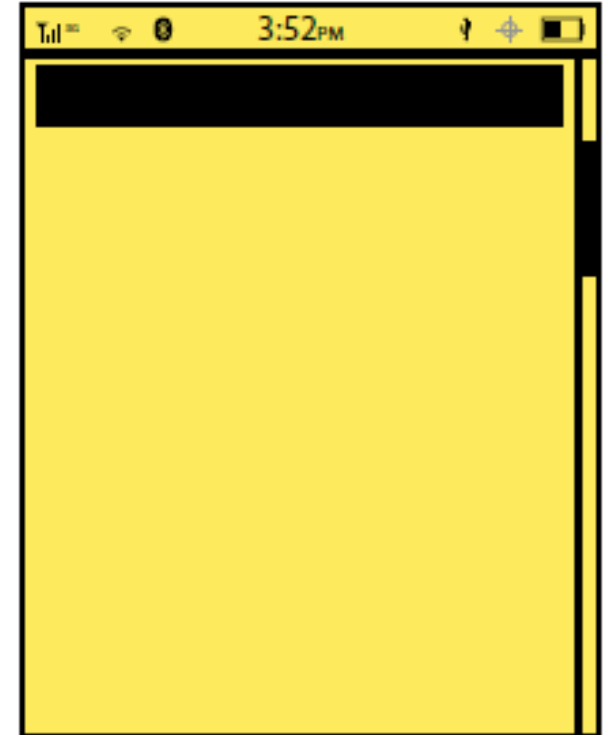
- Không được để quá nhiều vùng di chuyển, đặc biệt là không được phép scroll tới những vùng mà không có dữ liệu.
- Cân nhắc trong sử dụng scroll theo 2 trục. Nếu phải thể hiện scroll theo 2 trục, cần phải cung cấp đầy đủ và cần thiết hướng dẫn cho người dùng.
- Khi sử dụng lồng các kiểu scroll, tránh sử dụng cùng một loại scroll để lồng nhau.
- Khi sử dụng chỉ mục trong scroll, cần phải thể hiện chỉ mục đó đủ nhỏ để không ảnh hưởng tới thông tin của trang giao diện.
- Tránh mất thông tin khi sử dụng scroll

Dòng thông báo

- Mục đích – Cung cấp các trạng thái quan trọng liên quan tới phần cứng, thông tin pin và kết nối mạng, GPS ...
- OS cung cấp chức năng về dòng thông báo, nhưng trong thiết kế có thể phân bổ lại và kết hợp thể hiện vào trong chương trình

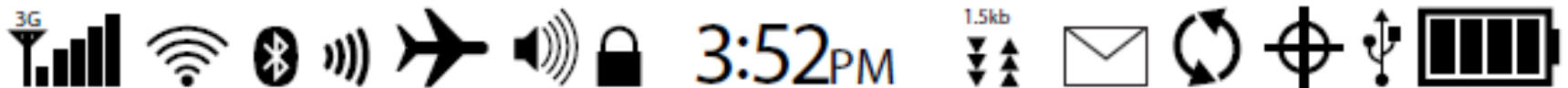
Dòng thông báo

- Được thể hiện ở phần trên của khung nhìn.
- Lưu ý, scroll sẽ dừng tại dòng thông báo (như là không có scroll)
- Dòng thông báo phải thu hút người sử dụng thông qua âm thanh, màu sắc nhẹ ...



Dòng thông báo

- Được thể hiện ở tất cả các màn hình.
- Được thể hiện trong khoảng thời gian nhất định và sẽ biến mất khi chức năng chính của chương trình được sử dụng.

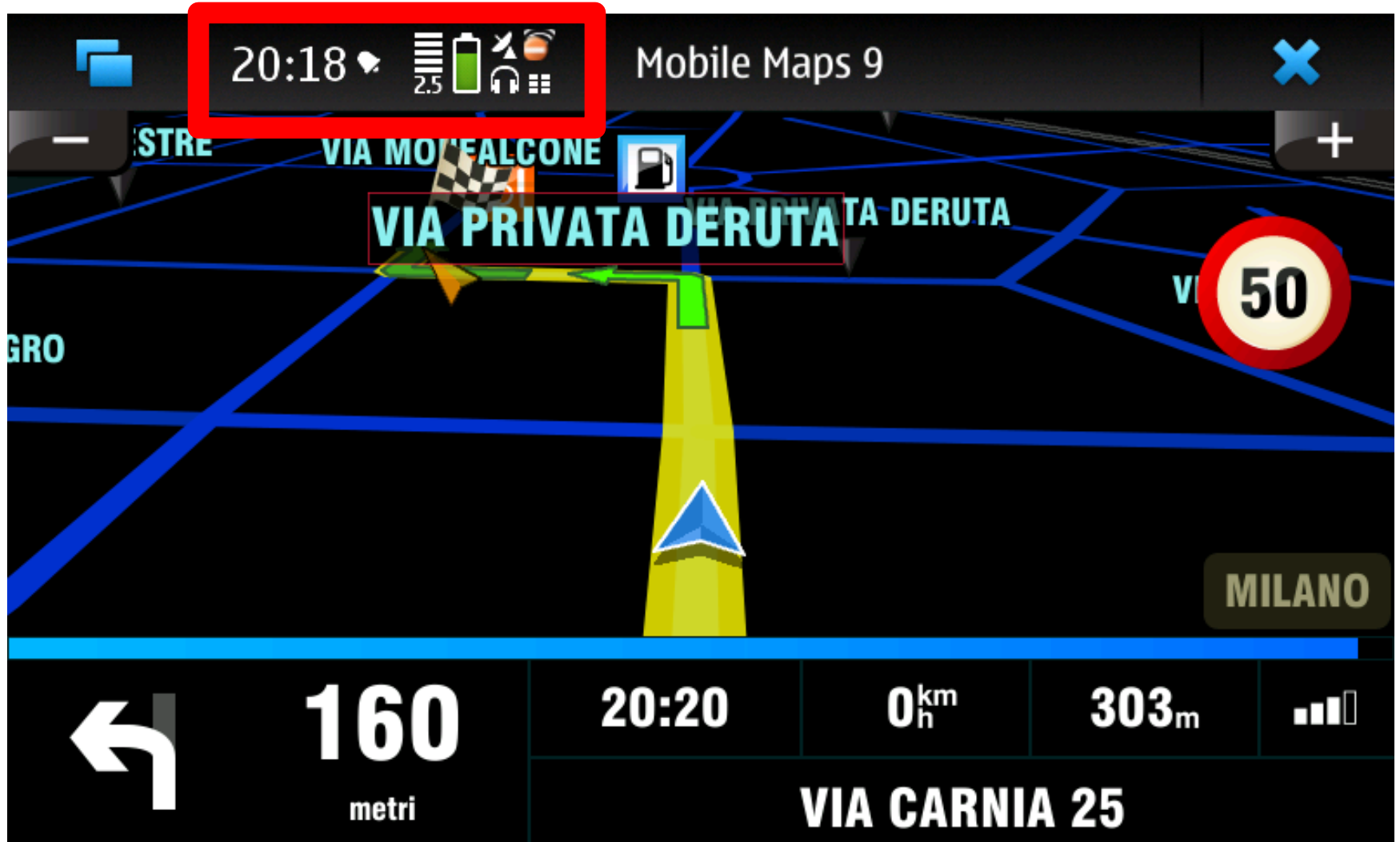


Dòng thông báo

■ Chú ý khi thiết kế:

- Luôn giữ nguyên trật tự và kích thước tại các màn hình khác nhau
- Không sáng tạo lại các thứ đã mặc định. Tái sử dụng luôn là một hình thức tốt cho một thiết kế tốt.
- Trừ khi thông báo cho người dùng các điều kiện sử dụng đặc biệt (pin trong màn hình camera), không được chọn thông báo nào được sử dụng.
- Tránh sử dụng các animation trong các dòng thông báo

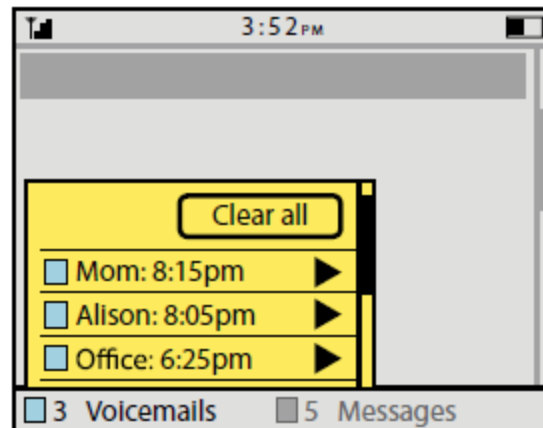
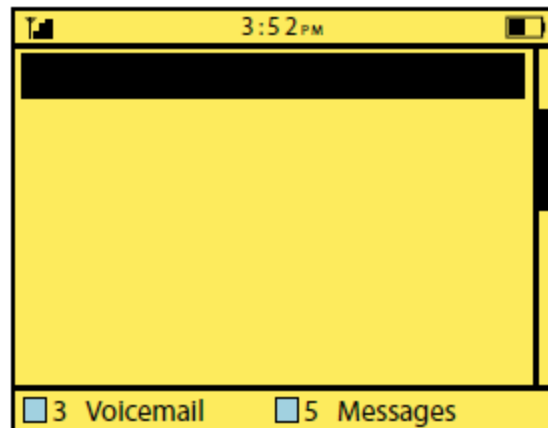
Ví dụ



Notifications

- Mục đích – sử dụng để thông báo tới người dùng về các trạng thái sử dụng chương trình, phòng tránh, lỗi ...
- Do OS cung cấp sẵn các thư viện

Notifications

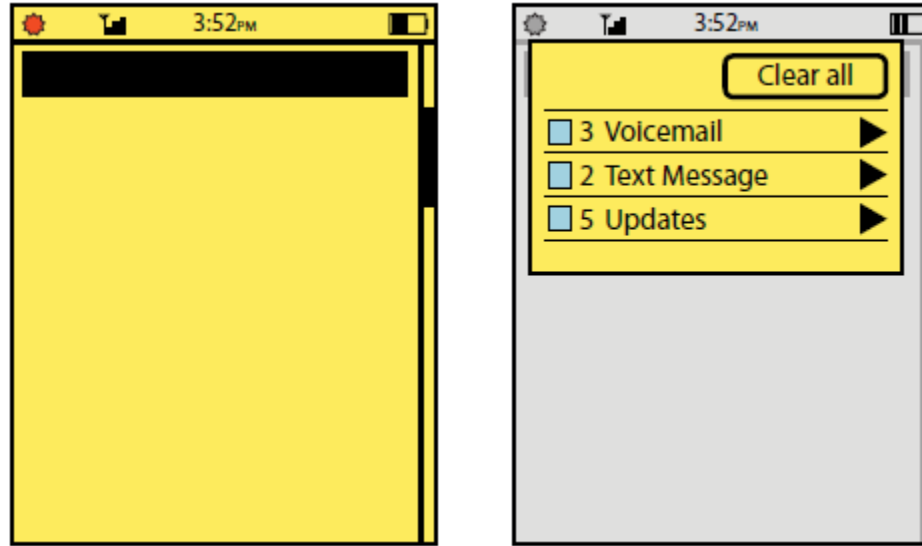


- Các vùng thông báo thường được đặt ở các vị trí phía trên hoặc dưới của màn hình, và không ảnh hưởng tới nội dung hiển thị.
- Thông báo phù hợp khi người sử dụng muốn lựa chọn các thông tin khác nhau thông qua tương tác vào thông báo

Notifications

- Có thể thể hiện nhiều thông báo trên một màn hình.
- Khi thông báo được mặc định là tập con của thiết bị và được sử dụng trong chương trình, hãy đảm bảo nó có cùng nguyên tắc với các thông báo của chương trình, và không xung đột với thông báo của OS.
- Thông báo có thể kết hợp với dòng thông báo (đã nói ở trên) như trong thông báo về sms, email ...

Notifications

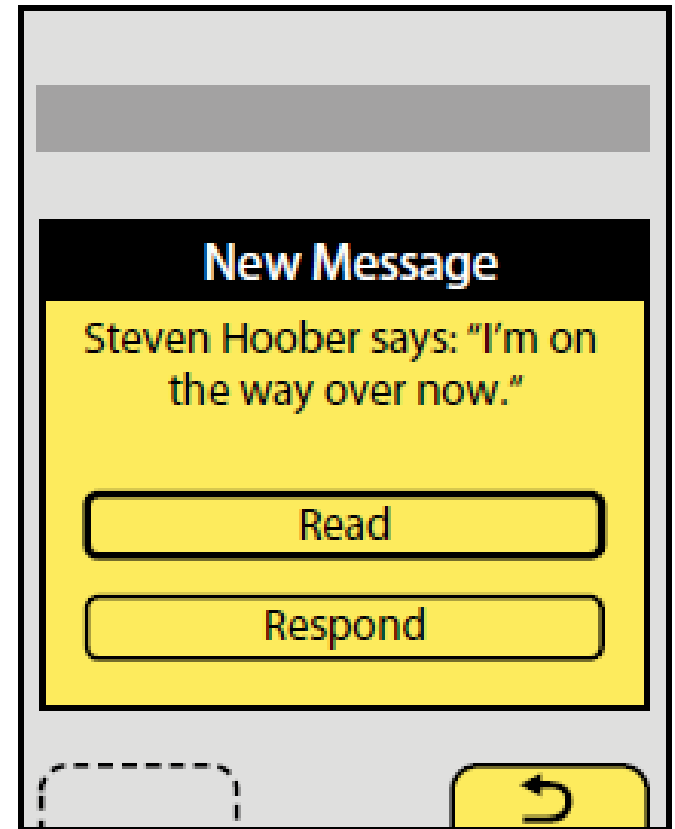


- Ví dụ dòng thông báo ở phía trên màn hình, có thể kết hợp với vùng thông báo khi người dùng lựa chọn.

Notifications

■ Ràng buộc:

- Người dùng phải nhìn thấy được các thông báo.
- Khi có nhiều thông báo cần đảm bảo các thông báo đó được hiển thị
- Khi một thông báo được lựa chọn, phải đảm bảo được đúng chức năng và thông tin liên quan tới thông báo đó



Thêm về thông báo

- Thông báo có thể sử dụng các phương pháp sau để tương tác với người dùng:
 - Sử dụng âm thanh
 - Sử dụng tín hiệu LED mà thiết bị có hỗ trợ
 - Sử dụng các hiệu ứng đối với dòng, màu sắc hiển thị thông báo

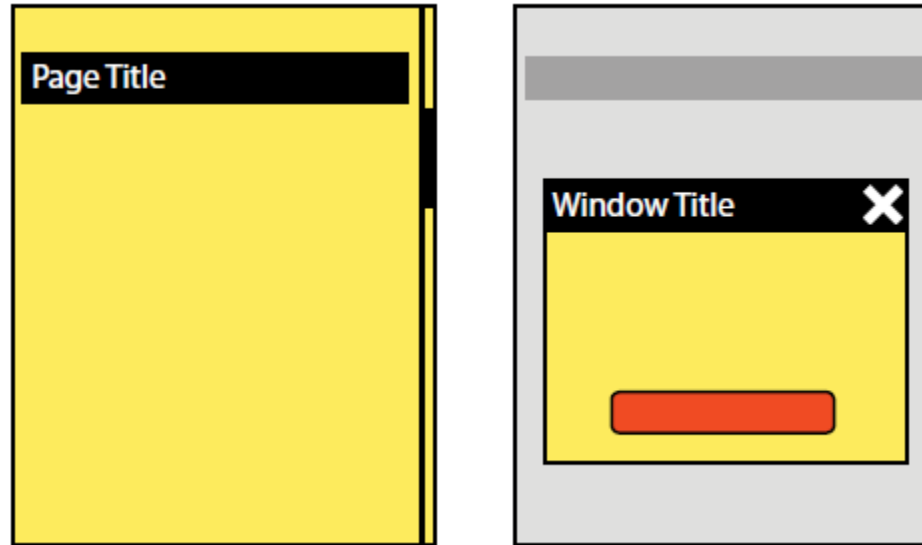
Các việc cần tránh trong thông báo

- Không được hiển thị thông báo nối tiếp nhau. Nếu có nhiều thông báo tại một thời điểm, cần phải sử dụng phương pháp thông báo theo nhiều dòng.
- Không được để thông báo ảnh hưởng tới các chức năng khác của hệ thống và chương trình.
- Không được thể hiện thông báo khi nối thiết bị với màn hình ngoài như sử dụng projector, màn hình TV ...

Tiêu đề

- Mục đích – Sử dụng gắn nhãn cho chương trình, cho các màn hình, hoặc thể hiện nội dung, hoặc tiến trình đã hoạt động.
- Do OS cung cấp các thư viện

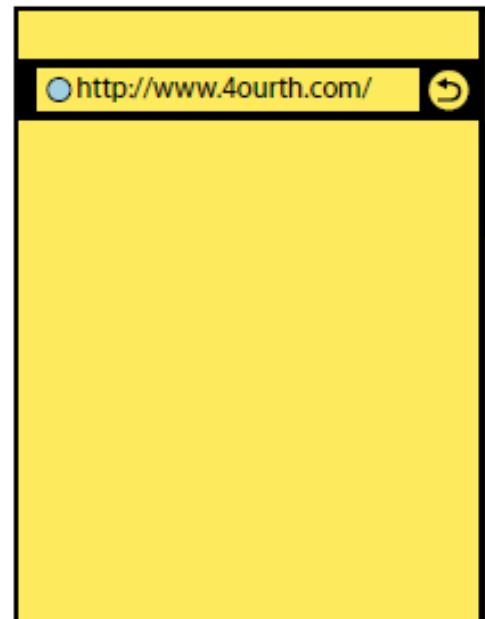
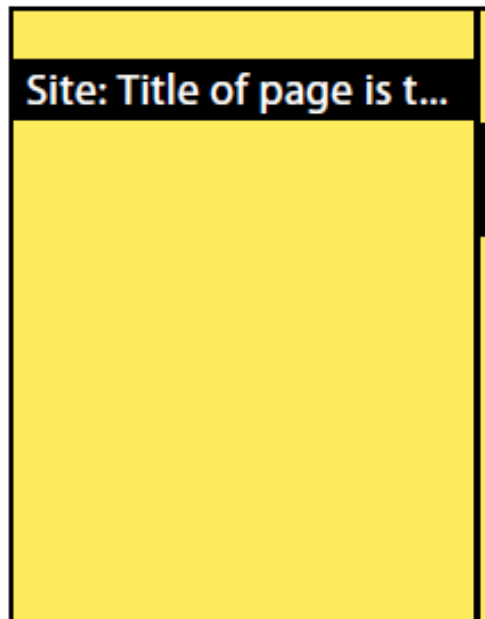
Tiêu đề



- Được thể hiện dưới dạng một phần tử đứng riêng như trên giao diện, cửa sổ, pop-up ...
- Tiêu đề thường được hiển thị theo chiều ngang.
- Thiết kế quan tâm tới vị trí, kích cỡ, nội dung và kiểu của tiêu đề

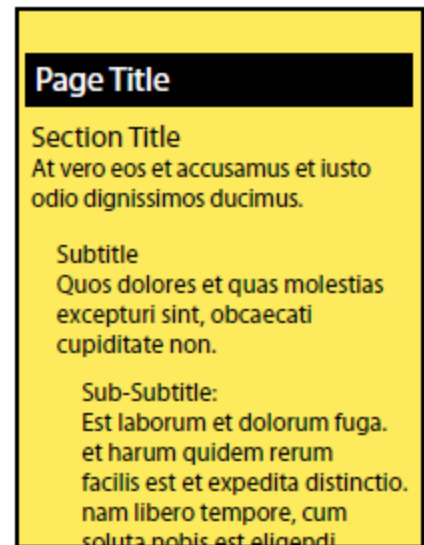
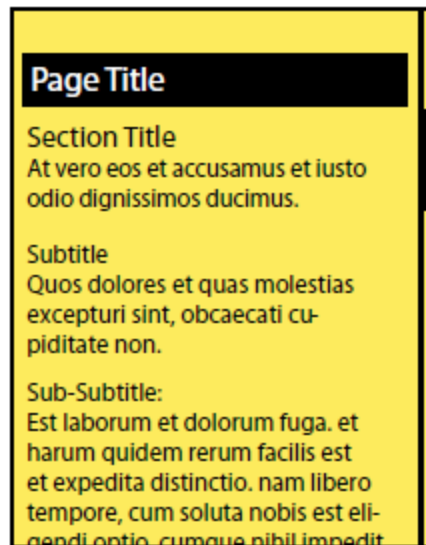
Vấn đề tương tác với tiêu đề

- Tiêu đề không nhất thiết phải có bất kỳ tương tác nào.
- Tương tác có thể có với tiêu đề là tương tác để tạo link trong tiêu đề khi sử dụng với giao diện dạng web.



Nội dung thể hiện tiêu đề

- Tiêu đề có thể bao gồm icon
- Sử dụng ngôn ngữ đồng nhất cho tiêu đề.
- Thống nhất về kích cỡ chữ, kiểu in hoa.
- Khi tiêu đề dưới dạng link, thể hiện rõ ràng, theo ràng buộc nguyên tắc được sử dụng xuyên suốt trong site hoặc chương trình.



Việc cần tránh trong tiêu đề

- Tránh dùng biệt ngữ trong tiêu đề.
- Tránh sử dụng từ ngữ thô, hoặc từ ngữ có thể gây khó hiểu cho người dùng.
- Không được dùng lặp lại nội dung tiêu đề. Nếu chương trình được mô tả đầy đủ, không để chương trình khởi động tại tên trong các trang con.

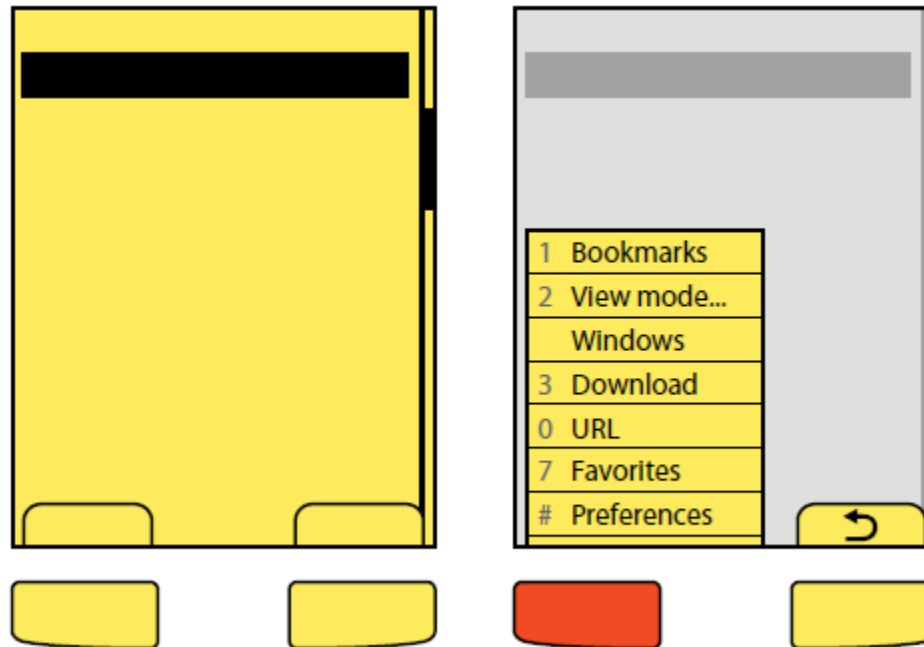
Các dạng menu

- Có 2 dạng chính:
 - Menu trong quan hệ
 - Menu cố định

Menu trong quan hệ

- Mục đích – sử dụng để hiển thị các chức năng khác của chương trình khi mà một màn hình không thể hiển thị được hết.
- OS cung cấp các kiểu chung nhất của cấu trúc menu. Người thiết kế phải định nghĩa các thành phần và các kiểu menu được sử dụng.

Menu trong quan hệ



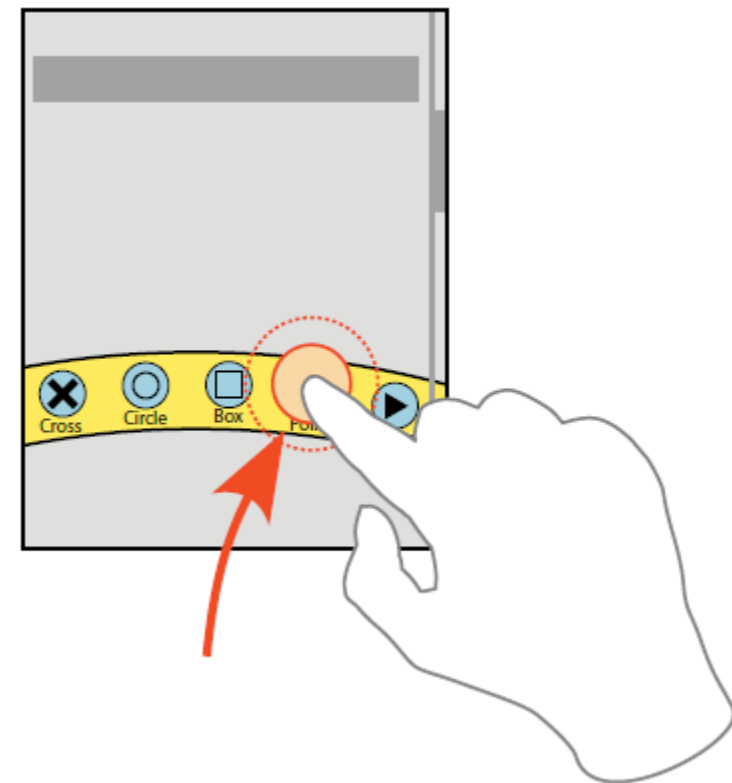
- Khi người dùng lựa chọn, một cửa sổ nhỏ hiển thị ra, chứa các chức năng liên quan (quan hệ) tới nội dung của chương trình

Menu trong quan hệ

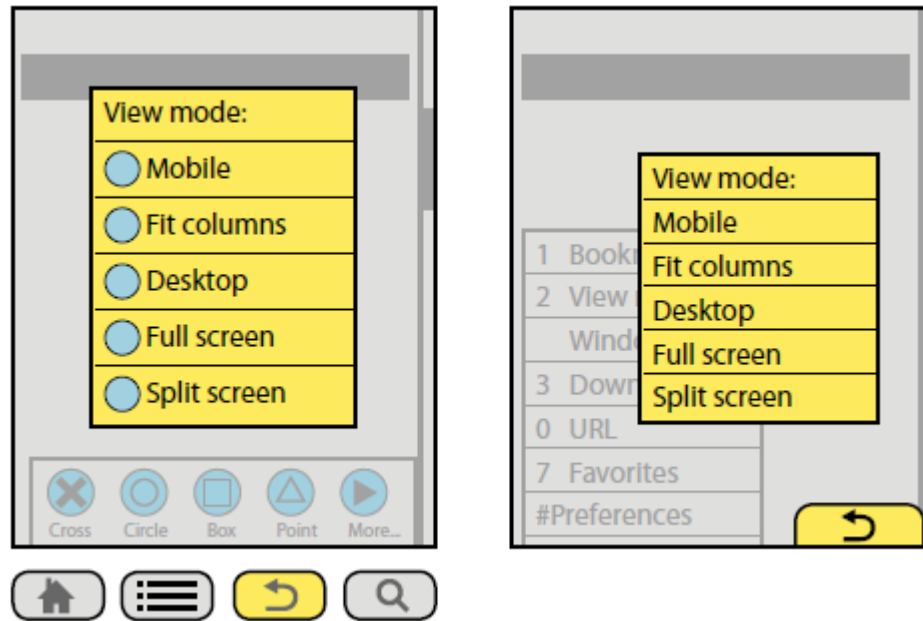
- Hình thức thiết kế menu cần phải liên quan chặt chẽ tới thiết bị:
 - Thiết bị có khả năng cung cấp một số phím cứng để kích hoạt menu.
 - Thiết bị cũng cung cấp một số phím sử dụng để quay lại các màn hình ...

Menu trong quan hệ

- Menu có thể khác nhau, nhưng phải thống nhất trong kiểu tương tác.
- Là tổ hợp giữa các phím do thiết bị cung cấp, cùng các phím lựa chọn menu trên chương trình.
- Có thể lựa chọn menu thông qua cử chỉ



Một số vấn đề về menu và menu con



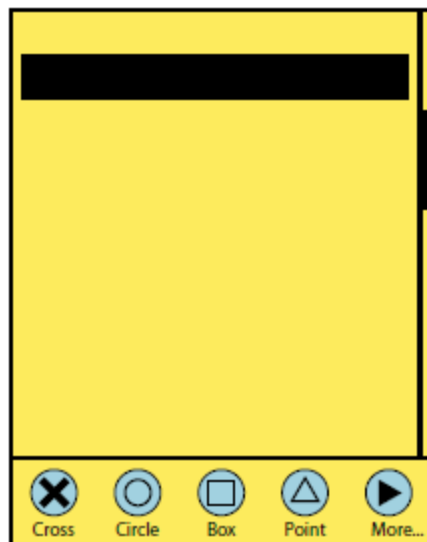
- Menu con sử dụng để lựa chọn thêm các chức năng con.
- Nó có thể là menu qua icon, cử chỉ, thường là được mở bằng một dialog thẳng riêng

Tránh trong sử dụng menu trong quan hệ

- Không cho phép tất cả chức năng có thể lựa chọn được, chỉ cho phép chức năng mà có thể được sử dụng bởi người dùng.
- Tránh sử dụng quá nhiều cấp menu khác nhau. Chỉ nên sử dụng 1 mức menu nếu cần.

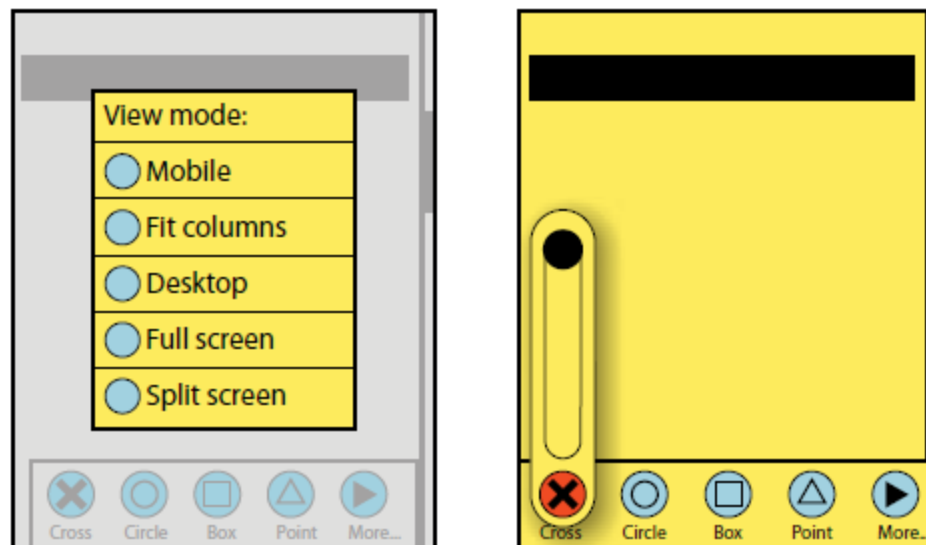
Menu cố định

- Mục đích – là các menu luôn hiển thị trong chương trình, được sử dụng để truy cập vào các chức năng hoặc vào các menu khác trong quan hệ.
- Người thiết kế phải đưa ra các kiểu, tính thống nhất trong menu cố định và kết hợp với phím cứng của thiết bị



Ý nghĩa của menu cố định

- Được sử dụng để thể hiện danh sách các chức năng có thể có của chương trình.
- Có thể bao gồm toàn bộ chức năng, hoặc các điều khiển cần thiết, và được kết hợp với menu trong quan hệ phù hợp với các hình thức view khác nhau.



Các vấn đề cần tránh trong menu cố định

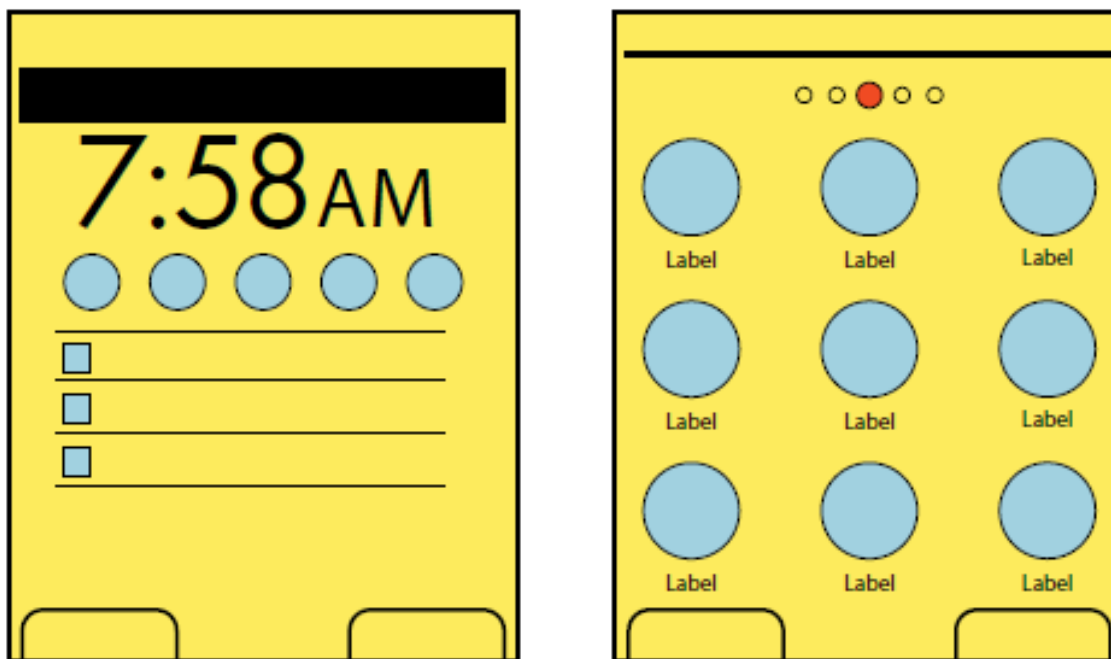
- Không được chồng các menu cố định.
- Tránh sử dụng menu cố định trên các thông báo.
- Tránh để menu cố định vi phạm vào các vùng hiển thị khác, ví dụ như vùng hiển thị của thông báo.

Một số vấn đề về màn hình

- Các kiểu màn hình
 - Màn hình chính và màn hình nghỉ
 - Màn hình khóa

Màn hình chính và nghỉ

- Thể hiện màn hình mặc định của thiết bị, chương trình khi được khởi động.
- Được thể hiện qua sự cung cấp thư viện của OS như widgets



Các biến thể

- Hầu hết các thiết bị đều có một số các phương pháp thiết kế màn hình khác nhau.
- Màn hình nghỉ (idle) là màn hình đơn được kích hoạt khi thiết bị khởi động, hoặc khi thoát khỏi chương trình.
- Màn hình chính chứa các mức menu khác nhau mà liên kết với các chương trình.
- Màn hình nghỉ có 2 mẫu:
 - Chỉ hiển thị trạng thái, mà không cho kích hoạt chương trình trực tiếp
 - Là liên kết giữa các màn hình khác nhau dựa trên icon

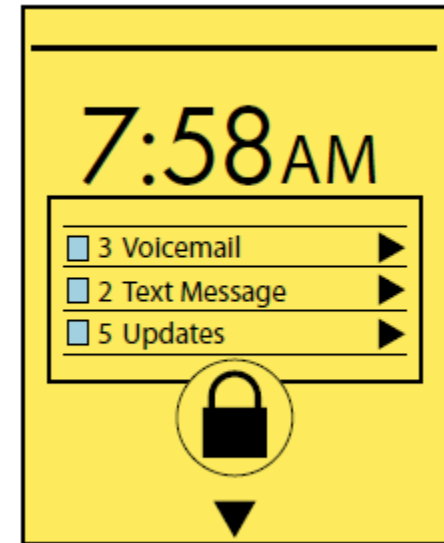
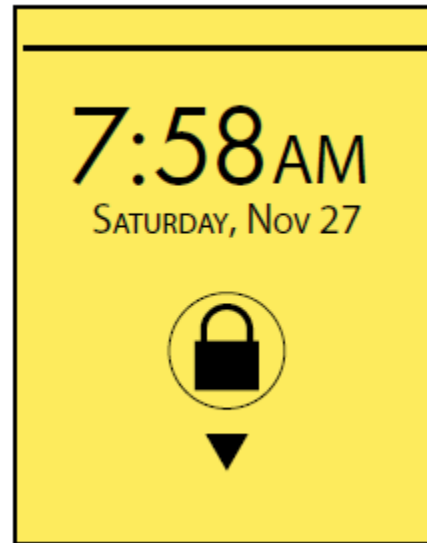
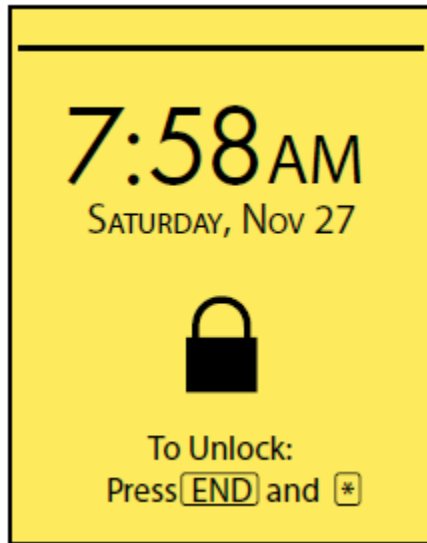
Cần quan tâm

- Chắc chắn là người dùng có thể hiểu về mô hình của màn hình chính và màn hình nghỉ mà không có sự huấn luyện, hỗ trợ.
- Tránh vi phạm vào mô hình UI của thiết bị dành cho màn hình chính và nghỉ.
- Phân biệt rõ ràng khi thiết kế màn hình.

Màn hình khóa

- Mục đích – được sử dụng để giảm mát mát năng lượng, bảo vệ thông tin người dùng nhập vào, phòng tránh các trường hợp nhập tin không cho phép.
- OS cung cấp màn hình khóa mặc định, nhưng hoàn toàn có thể can thiệp lại thông qua thiết kế, và xây dựng chương trình.

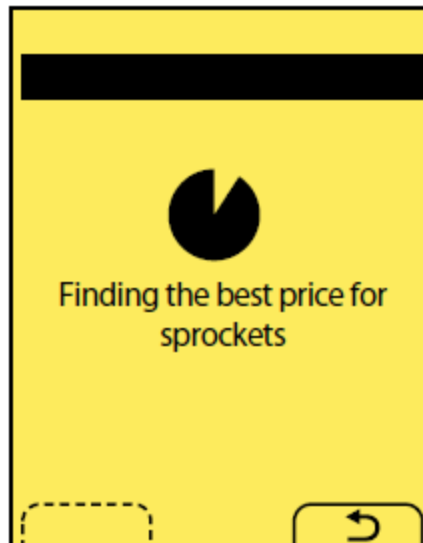
Màn hình khóa



- Được sử dụng khi thiết bị bị khóa hoặc ngủ.
- Có thể đưa các thông tin ra ngoài màn hình khóa như các event, alert, thời gian, hướng dẫn mở khóa.

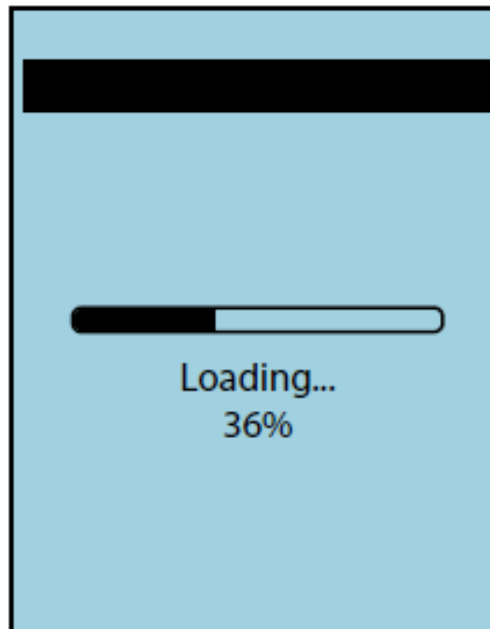
Màn hình chờ (màn hình xen kẽ)

- Mục đích – Hiển thị giữa các tiến trình khác nhau khi quá trình đợi chiếm một khoảng thời gian nào đó
- Màn hình chờ hữu ích trong thiết kế và xây dựng chương trình, nhằm giảm sự chờ đợi của người dùng.



Màn hình chờ

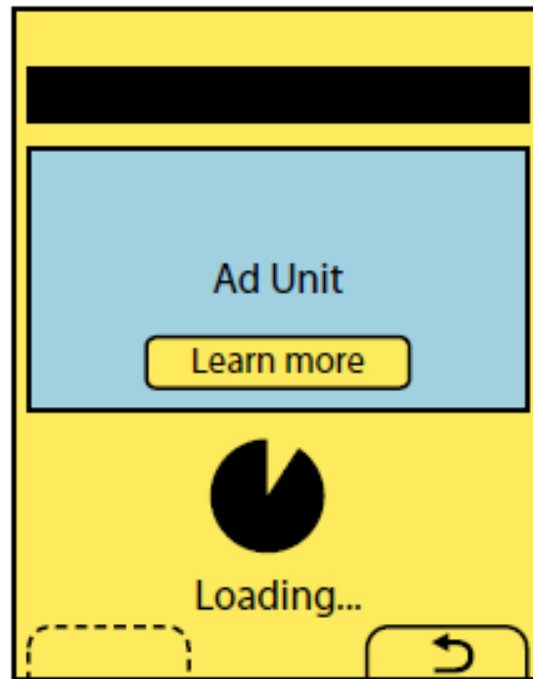
- Không có giới hạn kỹ thuật nào cho màn hình chờ.
- Được thể hiện trong chương trình, site, hoặc giữa các tiến trình. Nội dung hiển thị cần thiết phải rõ ràng, và đủ thông tin.



Kiểu màn hình chờ

■ Có 2 kiểu:

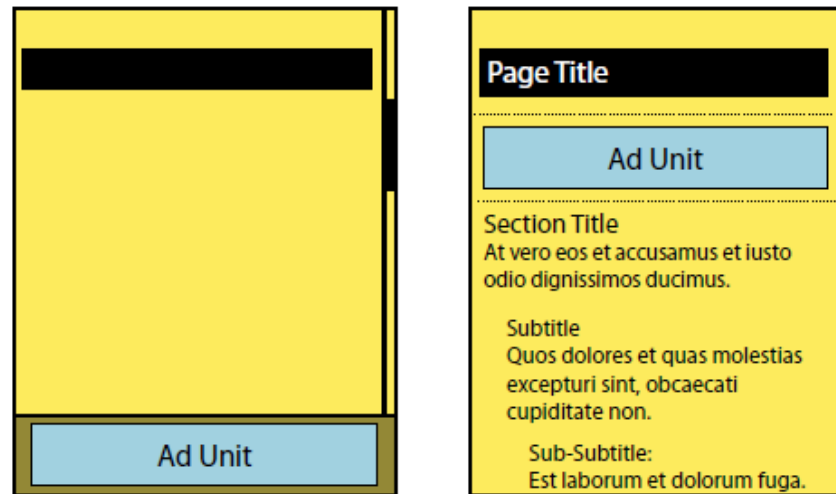
- Màn hình nạp (khởi động) chương trình.
- Màn hình xen kẽ giữa các tiến trình, chức năng trong sử dụng chương trình.



Quảng cáo

- Quảng cáo được thể hiện trong chương trình, site hoặc các service khác.
- Quảng cáo được gắn với chương trình như là một layout riêng biệt.
- Phải không được ảnh hưởng tới các chức năng chính của chương trình
- Không ảnh hưởng tới sự phân bố giao diện cho chương trình chính

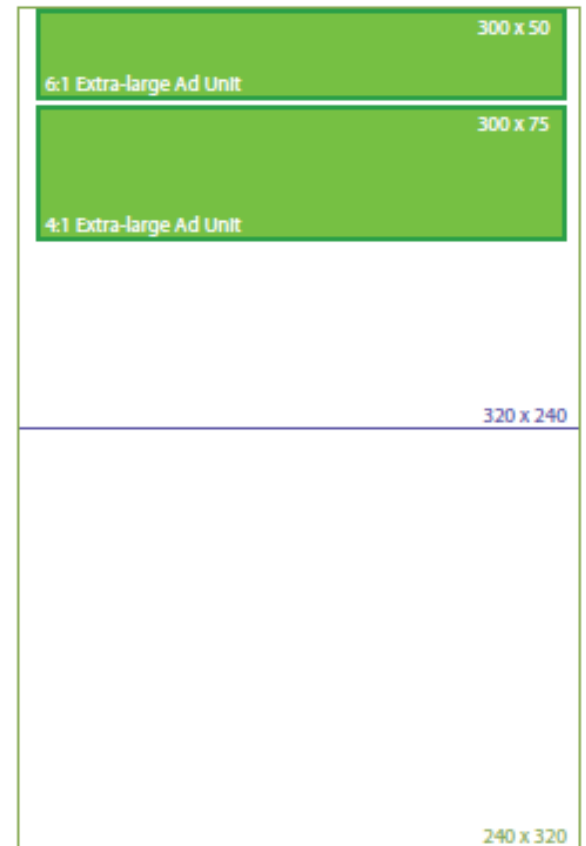
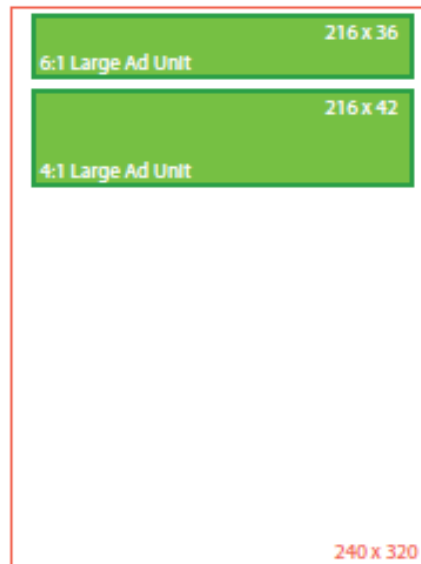
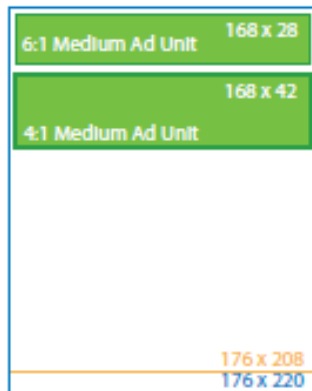
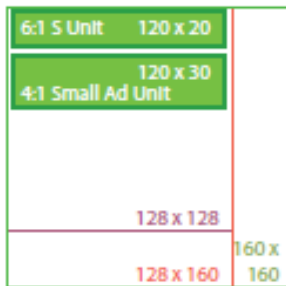
Quảng cáo



- Quảng cáo có thể chiếm ở trong trang view, hoặc ở ngoài trang view.
- Quảng cáo phải:
 - Rõ ràng, dễ đọc, hiểu
 - Dễ tương tác
 - Sử dụng thống nhất tại 1 vị trí đối với tất cả các màn hình chương trình.
 - Đủ nhỏ để không ảnh hưởng tới layout khác của chương trình

Hình thức thể hiện

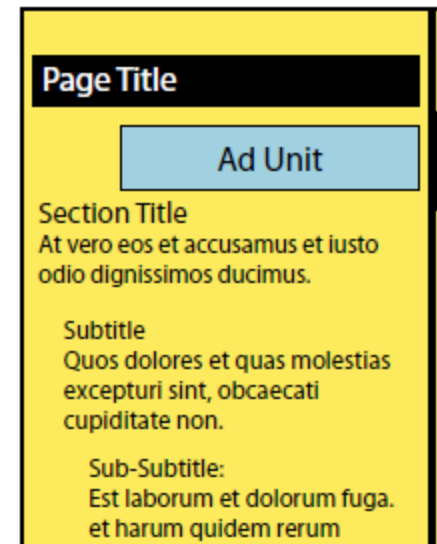
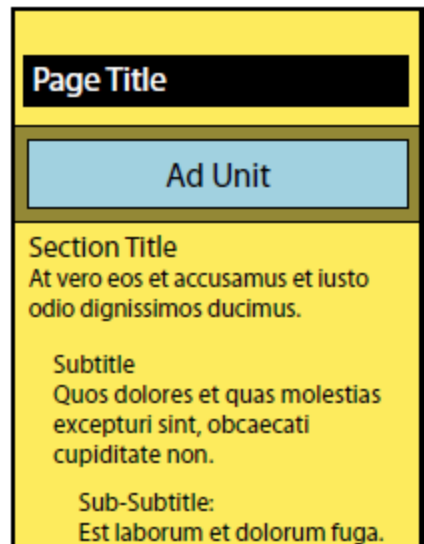
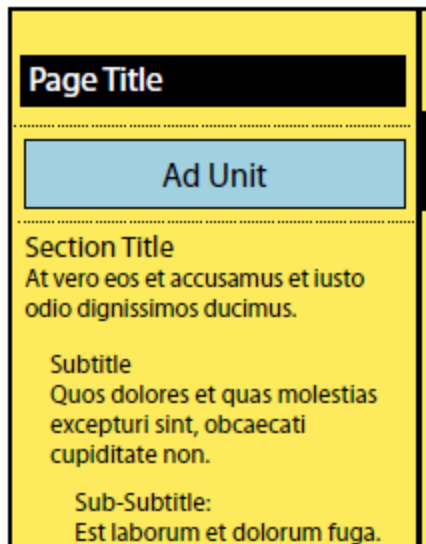
- Lấy từ tiêu chuẩn MMA về banner quảng cáo



Phương pháp thể hiện quảng cáo

■ Có 3 phương pháp thể hiện:

- Thể hiện dưới dạng bóng mờ, hoặc được phủ màu theo bề ngang của giao diện màn hình.
- Tách biệt với các nội dung còn lại
- Nếu quảng cáo nhỏ hơn màn hình view, sử dụng cách sắp xếp khác nhau.



Tổng kết

- Thông qua bài này, cần phải nắm:
 - Các hình thức thể hiện trên giao diện mobile.
 - Tập trung đặc tả yêu cầu người dùng, các chức năng, mục tiêu và ngữ cảnh sử dụng trong thiết kế
 - Nguyên tắc thể hiện các trang layout, cấu trúc nội dung thể hiện trên layout

Bài tập số 3

- Xây dựng thiết kế các màn hình chi tiết, và lập trình với nội dung:
 - Chương trình chạy trên Android, sử dụng để hiển thị lại nội dung có trên trang web sinh viên của trường, với trình tự cơ bản: mỗi sinh viên có 01 account đăng nhập vào hệ thống, hiển thị các thông tin chung và riêng, các thao tác với dữ liệu và account, và thoát khỏi hệ thống.
 - Hướng dẫn: sẽ có 2 hệ thống – Backend là CSDL thu thập được từ trang web trên; và Frontend là chương trình trên mobile.
 - Có thể sử dụng 1 trong các mã nguồn sau (khuyến khích xây dựng trên nền web):
 - <http://phonegap.com/>
 - <https://www.djangoproject.com/>

Bài tập số 3

- Các tài liệu cần nộp:
 - Tài liệu thiết kết các màn hình chi tiết
 - Tài liệu báo cáo chương trình
- Source code chương trình + comment các chức năng chính.
- Thời gian làm bài trong 4 tuần tới 5/11 (với INT3115-3) và 8/11 (với INT3113-2)