Aston University
BIRMINGHAM UK

CS3ID Interaction Design

Dr Lilit Hakobyan
Dr Abinaya Sowriraghavan
Dr Marwa Gadala

UNIVERSITY OF THE YEAR 2020 The Guardian

learning outcomes…

- at the end of the two lectures on Design Patterns (together with additional reading), you should be able to:
  - recognise and describe different application structure patterns and be able to recommend and apply such patterns in UI design based on application characteristics
  - understand the basics of navigation and appreciate the costs of navigation
  - recognise and describe different navigational patterns and be able to recommend and apply navigational patterns in UI design based on application characteristics
  - recognise and describe different page layout patterns and be able to recommend and apply such patterns in UI design based on application characteristics
  - recognise and describe different action and command patterns and be able to recommend and apply such patterns in UI design based on application characteristics

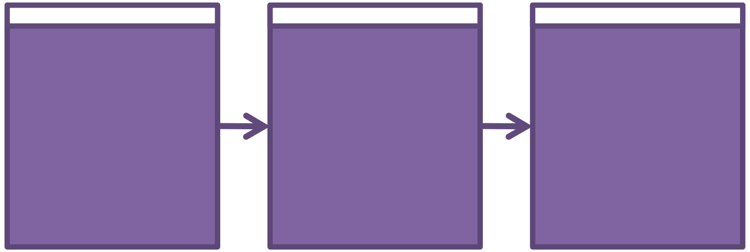
- **strongly recommended that you read the additional reading for this material to cover examples we won't have time for here!**

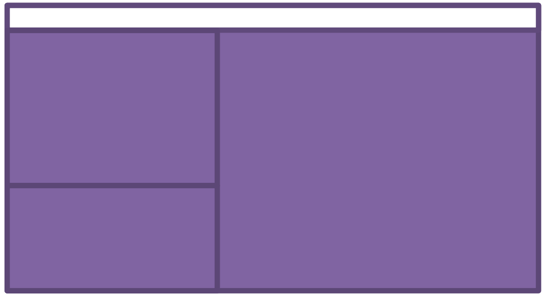
basics of information architecture…

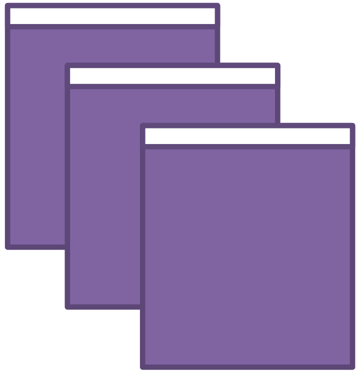
- figure out how to structure all your content and functionality
  - how to organise it, label it, and guide a user through the UI
- need to structure the UI so users know what to do next (or at least have a good idea where to look!)
- base decisions on:
  - nature and domain of the application
  - users' domain knowledge
  - users' comfort levels with computers
  - how closely your application needs to match the users' mental models of the domain

basics of physical structure…

- have to translate a design into a physical structure of windows, pages, and controls
- should an application use multiple windows, a single window with several tiled panes, or one window with swapping-out content?
- technology sometimes restricts options – e.g., mobile phones
- analyse tasks users will perform
  - do they need to work in 2+ areas at same time?

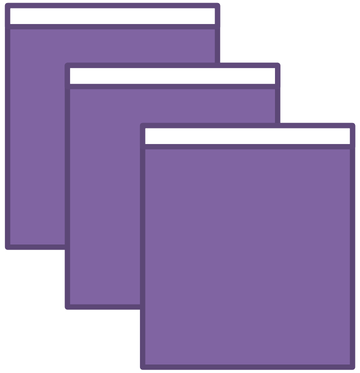

one-window paging


tiled panes


multiple windows

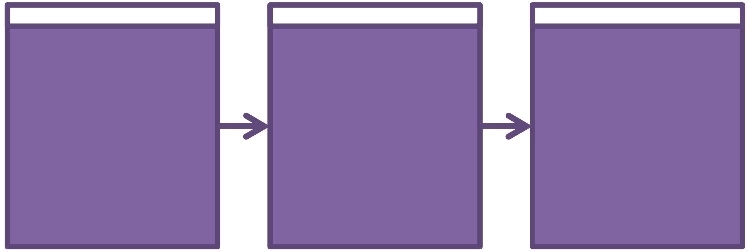
multiple windows…

- sometimes the right choice, but not often
  - work best in applications where users want to customise their screen
- users (especially infrequent users) tend to find multiple windows irritating or confusing
  - lose them if too many onscreen at once
- need to use multiple windows or tiled panes if users need to see 2+ windows in parallel



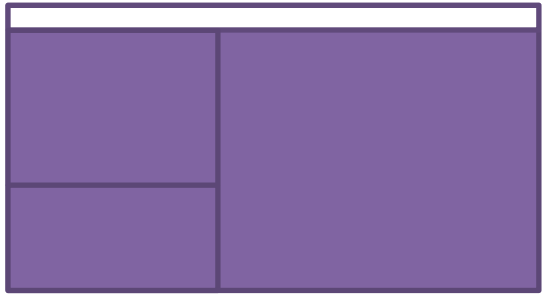
one-window paging…

- simple web applications work best with a single-window paging model – shows one page at a time
    - people are very familiar with this model
- conserves space
    - nothing onscreen to compete with window content
    - best choice for small devices



tiled panes…

- great for users who want to see a lot at once while expending little effort on window management
  - 2- and even 3-pane structures are common
- can take up a lot of screen space
- tiled panes and multiple windows = “open floor plan” concept
  - i.e., expose large number of features in one place
  - users are responsible for focussing their own attention to the right panel and right feature at the right time
  - best not to use these if you need to lead a user along a specified path
- choosing a physical structure is all about trade-offs!



application structure patterns…

- patterns known to work very well:
  - two-panel selector
  - canvas + palette
  - one-window drilldown
- patterns that focus on content in the abstract:
  - wizard (linearising a path through a task)
  - extras on demand* } additional ways to divide up content
  - intriguing branches* }
- ways to integrate help directly into an application:
  - multi-level help*

*** read Tidwell, J., (2005), Designing Interfaces, O'Reilly Media Inc. Chapter 2**

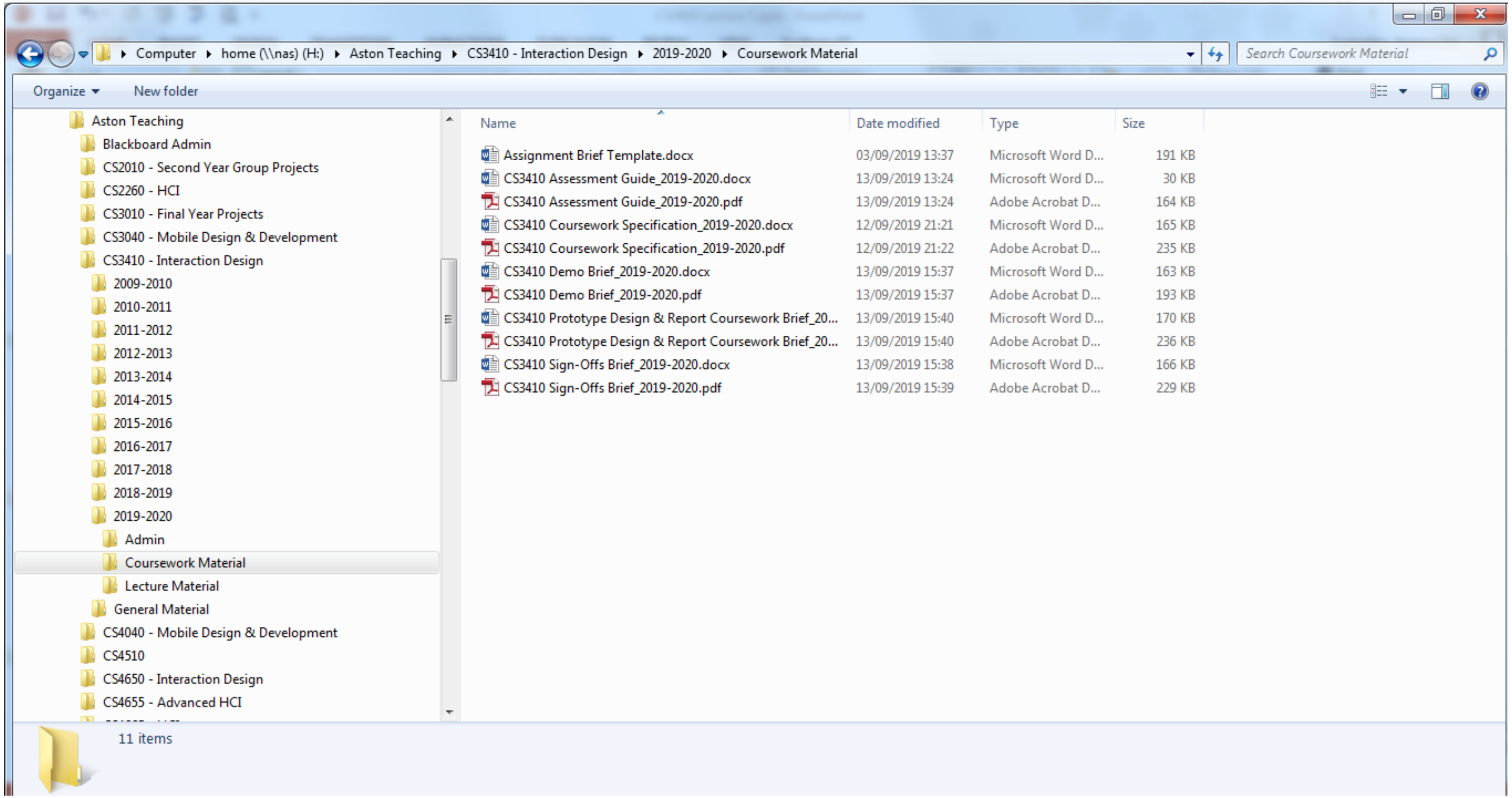
two-panel selector pattern…

- 2 side-by-side panels
  - one shows set of items user can select » other shows content of the selected item
- use when presenting a list of objects, categories, or actions
  - e.g., messages in a mailbox, images in a library, files, etc.
  - users should see structure of list but should be able to walk through the items at their own pace and in their chosen order
  - display you are working with has to be large enough to show separate panels at once (e.g., very small mobile phones can't cope with this)
- it is a learned convention but is common and powerful
  - users apply concept to applications that look similar to ones they are already familiar with (e.g., mail tool)

two-panel selector pattern contd…

- advantages:
  - users can quickly shift their attention between panels
  - reduces physical effort – visual focus is close and selection change only requires single mouse click/key press
  - reduces visual cognitive load – with other patterns user has to pay more attention to what he/she is looking at but when the window stays stable, the user can focus on the smaller area of the screen that changed
  - reduces memory load – information is constantly visible
- protocol:
  - place the selectable list at the top or left / place the details panel at the bottom or the right to take advantage of visual reading flow
  - when item is selected, immediately show the contents in the details panel
  - selection via single click (or key press)
  - make selected item visually obvious
  - use any of the list of objects models to display selectable list or other more complex structures (see Tidwell: Chapter 6 for more information on these)

two-panel selector example…



Computer ▸ home (\\nas) (H:) ▸ Aston Teaching ▸ CS3410 - Interaction Design ▸ 2019-2020 ▸ Coursework Material

Search Coursework Material

Organize ▾ New folder

- Aston Teaching
  - Blackboard Admin
  - CS2010 - Second Year Group Projects
  - CS2260 - HCI
  - CS3010 - Final Year Projects
  - CS3040 - Mobile Design & Development
  - CS3410 - Interaction Design
    - 2009-2010
    - 2010-2011
    - 2011-2012
    - 2012-2013
    - 2013-2014
    - 2014-2015
    - 2015-2016
    - 2016-2017
    - 2017-2018
    - 2018-2019
    - 2019-2020
      - Admin
      - Coursework Material
      - Lecture Material
    - General Material
  - CS4040 - Mobile Design & Development
  - CS4510
  - CS4650 - Interaction Design
  - CS4655 - Advanced HCI

Name	Date modified	Type	Size
Assignment Brief Template.docx	03/09/2019 13:37	Microsoft Word D...	191 KB
CS3410 Assessment Guide_2019-2020.docx	13/09/2019 13:24	Microsoft Word D...	30 KB
CS3410 Assessment Guide_2019-2020.pdf	13/09/2019 13:24	Adobe Acrobat D...	164 KB
CS3410 Coursework Specification_2019-2020.docx	12/09/2019 21:21	Microsoft Word D...	165 KB
CS3410 Coursework Specification_2019-2020.pdf	12/09/2019 21:22	Adobe Acrobat D...	235 KB
CS3410 Demo Brief_2019-2020.docx	13/09/2019 15:37	Microsoft Word D...	163 KB
CS3410 Demo Brief_2019-2020.pdf	13/09/2019 15:37	Adobe Acrobat D...	193 KB
CS3410 Prototype Design & Report Coursework Brief_20...	13/09/2019 15:40	Microsoft Word D...	170 KB
CS3410 Prototype Design & Report Coursework Brief_20...	13/09/2019 15:40	Adobe Acrobat D...	236 KB
CS3410 Sign-Offs Brief_2019-2020.docx	13/09/2019 15:38	Microsoft Word D...	166 KB
CS3410 Sign-Offs Brief_2019-2020.pdf	13/09/2019 15:39	Adobe Acrobat D...	229 KB

11 items

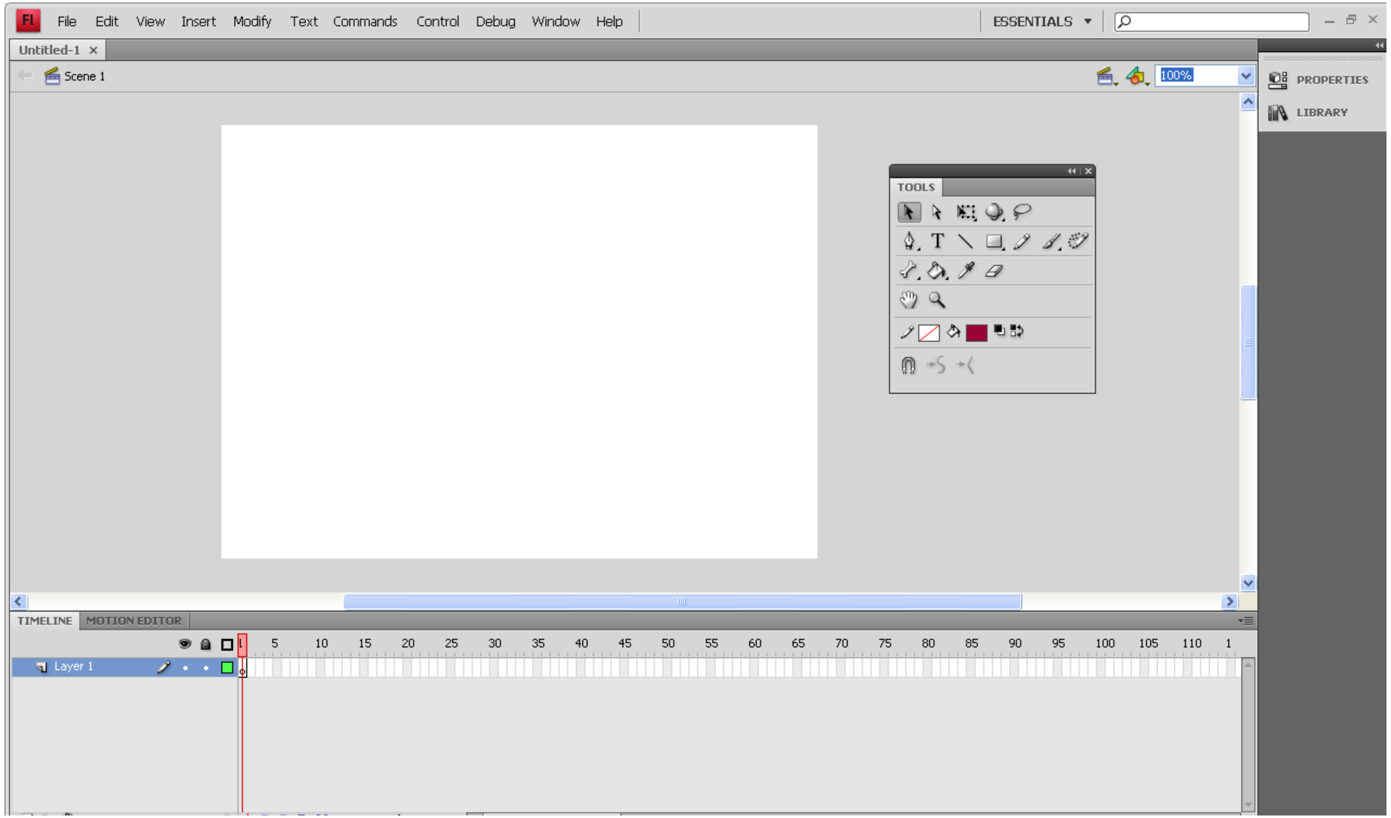
canvas + palette pattern…

- iconic palette next to a blank canvas
    - users click on the palette buttons to create objects on the canvas
- use when you are designing any kind of graphical editor
    - e.g., where use involves creating new objects and arranging them in some virtual space
- it is very common
    - natural mapping from familiar physical objects in the real world to the virtual, onscreen world
    - palette takes advantage of visual recognition (rather than recall)
    - often reuses iconic norms (e.g., paintbrush, hand, magnifying glass, etc.)

canvas + palette pattern contd…

- protocol:
  - present large, empty area as canvas
    - *might be in its own window, in a tiled window, or in a single pane with other tools*
    - *works as long as the user can view the canvas + palette simultaneously*
  - palette should be grid of iconic buttons (can include text if icons are too cryptic)
  - place palette to left or top of the canvas
    - *can divide into subgroups using a card stack (e.g., tabs) if necessary*
  - can use drag-and-drop to create the items on the canvas or can use a single click on the palette followed by a single click on the canvas

canvas + palette example…



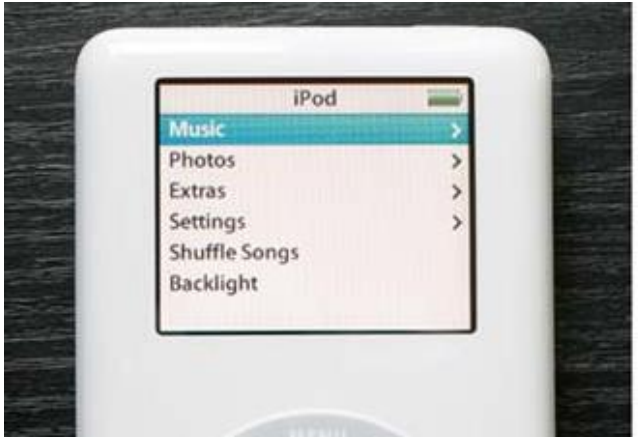
one-window drilldown pattern…

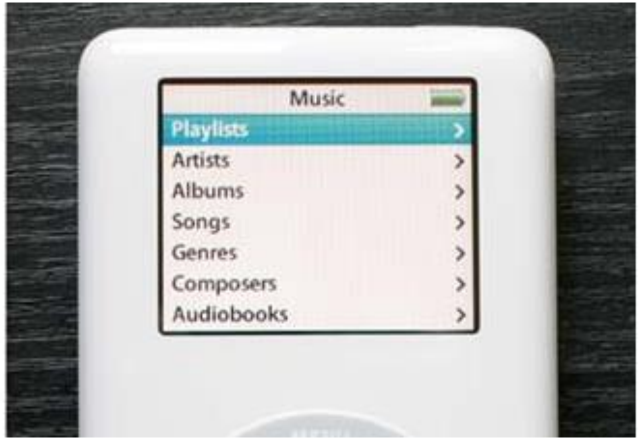
- show each of the application’s pages in a single window
    - as a user drills down through a menu of options or into an object’s details the window contents are completely replaced with the new content
- use when the application contains many pages of content which the user has to navigate through
    - e.g., address books, calendars, etc.
    - pages/content might be arranged linearly, in a hyperlinked network, in a menu hierarchy (most common)
    - use if you are building for devices with screen real estate restrictions or you have a complexity limit
- advantages of this pattern
    - keeps things simple – options at each stage are clear and users don’t need to focus attention anywhere else
    - common pattern (e.g., web browsers)

one-window drilldown pattern contd…

- protocol:
    - structure content into panels/windows that fit gracefully into the screen real estate
    - design obvious "doors" to other parts of the UI – e.g., buttons
    - when user clicks on button, replace the current panel with the appropriate new one
    - navigation either using back/forward buttons or a permanent toolbar
    - put "done" or "cancel" controls on panels where a user completes a task or activity
        - *sense of closure*
    - no graphical representation of application's structure – nor of location within that structure – so one-window drilldown forces users to rely on mental picture of how the content fits together
        - *simple linear or hierarchical structures work best*
        - *use breadcrumbs and sequence maps*

one-window drilldown example…





© Tidwell, J., (2005), *Designing Interfaces*, O'Reilly Media Inc.

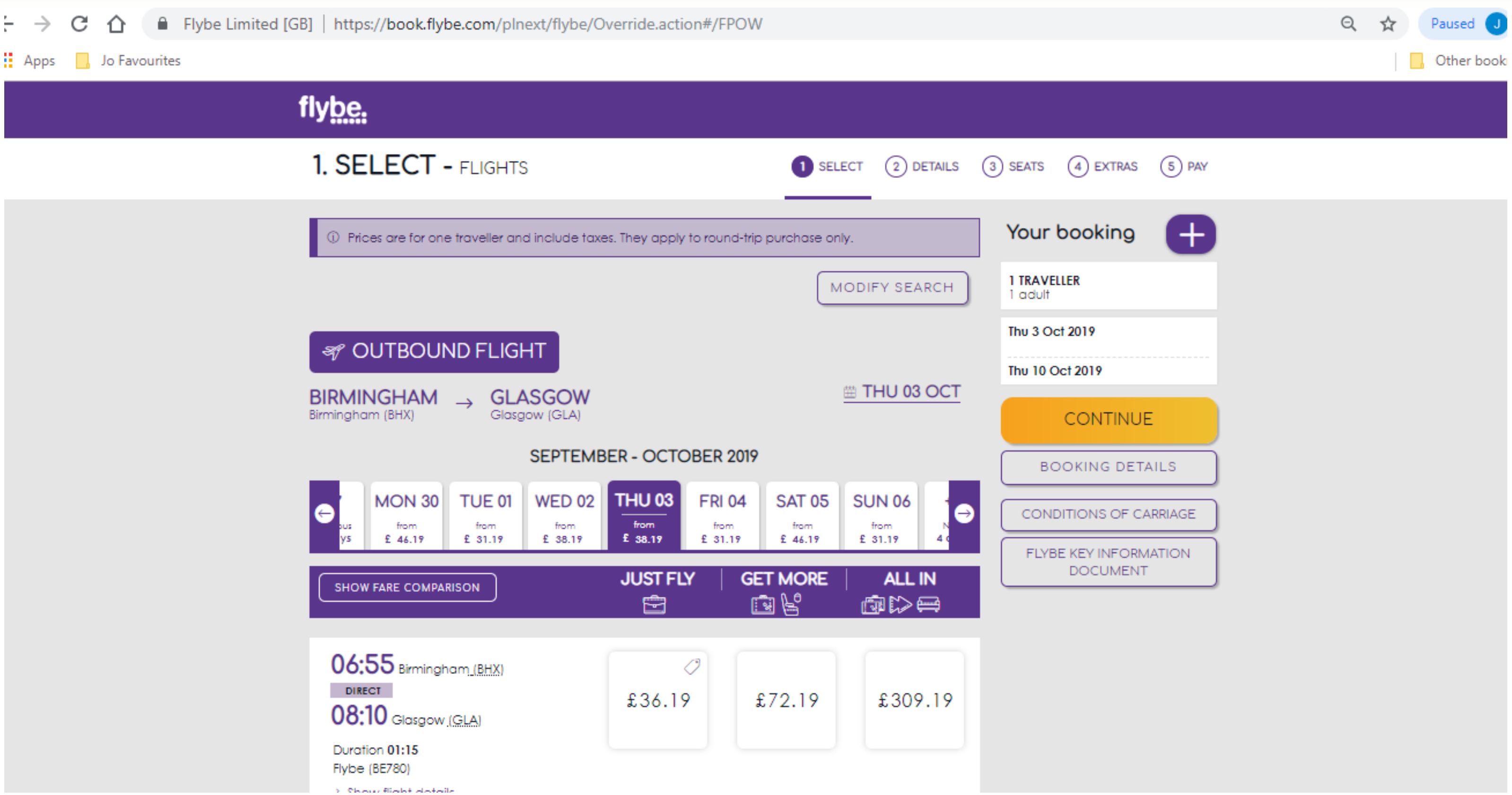
wizard pattern…

- leads the user through the UI step by step, doing tasks in a prescribed order
- use when the task is long or complicated or where the task is likely to be novel for the user
    - option where designers know more about the best way to get the task done than the user does
    - well-suited tasks include those with lots of user-made decisions that affect downstream choices
        - *user has to be willing to surrender control to the system*
        - *can be considered frustrating and inflexible*
- “divide and conquer”
    - chunking tasks to simplify process
    - save users effort of working out the task structure

wizard pattern contd…

- protocol:
  - chunk the task into groups of operations
    - *thematic breakdown*
    - *decision point-based breakdown*
  - hard to strike balance between too few and too many chunks
    - *each chunk shouldn’t be too large*
  - use one of the other patterns to display the chunks
    - *one-window drilldown is common*
    - *see layout patterns (Lecture 6 & additional reading)*
  - provide good/sensible defaults at each stage

wizard example…



Flybe Limited [GB] | https://book.flybe.com/plnext/flybe/Override.action#/FPOW

Apps Jo Favourites Other book

flybe.

1. SELECT - FLIGHTS

1 SELECT 2 DETAILS 3 SEATS 4 EXTRAS 5 PAY

ⓘ Prices are for one traveller and include taxes. They apply to round-trip purchase only.

MODIFY SEARCH

OUTBOUND FLIGHT

BIRMINGHAM → GLASGOW
Birmingham (BHX) Glasgow (GLA)

THU 03 OCT

SEPTEMBER - OCTOBER 2019

MON 30	TUE 01	WED 02	THU 03	FRI 04	SAT 05	SUN 06
from £ 46.19	from £ 31.19	from £ 38.19	from £ 38.19	from £ 31.19	from £ 46.19	from £ 31.19

SHOW FARE COMPARISON

	JUST FLY	GET MORE	ALL IN
06:55 Birmingham (BHX) DIRECT 08:10 Glasgow (GLA) Duration 01:15 Flybe (BE780)	£36.19	£72.19	£309.19

Your booking

1 TRAVELLER
1 adult

Thu 3 Oct 2019

Thu 10 Oct 2019

CONTINUE

BOOKING DETAILS

CONDITIONS OF CARRIAGE

FLYBE KEY INFORMATION DOCUMENT

navigation…

- make the "commute" as short or non-existent as possible!
- **signposts** – help users figure out their surroundings
  - e.g., titles, tabs, selection indicators, etc.
- **wayfinding** – process of finding way to target/goal
- features that help users when wayfinding:
  - **good signage** – clear, unambiguous labels that anticipate what users are looking for
    - *should be where users expect them to be and decision point options should help guide users to their end goal*
  - **environmental cues** – norms/conventions for button placement etc.
  - **maps** – provide mental picture of the whole application space

navigation contd…

- opening up an application window incurs a cognitive cost
  - **context switch** as users are forced to refocus their attention and adjust to their new surroundings – to familiarise themselves with the application layout, contents, exits, and work out how they are going to do what they need to do with the application
  - even if a user is already familiar with the application, as he/she context switches from another environment/application there is still a small cost
  - goal is to minimize the cognitive cost associated with context switching
- keep distances short
  - challenge: design application to do the most common 80% of use cases with 0 or one context switches
    - *avoid too many levels of nesting*
    - *but don’t sacrifice simplicity to achieve this!*

navigation/orientation patterns…

- patterns that describe navigational structures (how sections link):
  - clear entry points
  - global navigation
  - hub and spoke
  - pyramid
- patterns that focus on signposting current position:
  - sequence map
  - breadcrumbs
  - annotated scrollbar*
  - colour-coded sections*
- others:
  - animated transition* (trick to preserve users' sense of orientation)
  - escape hatch*

read Tidwell, J., (2005), Designing Interfaces, O'Reilly Media Inc. Chapter 3

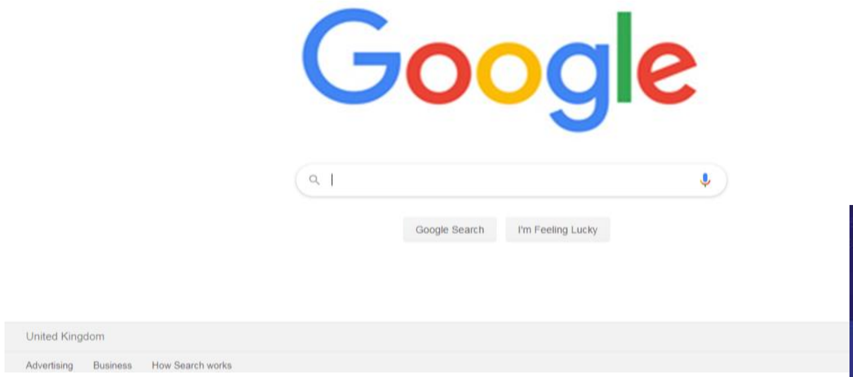
clear entry points pattern…

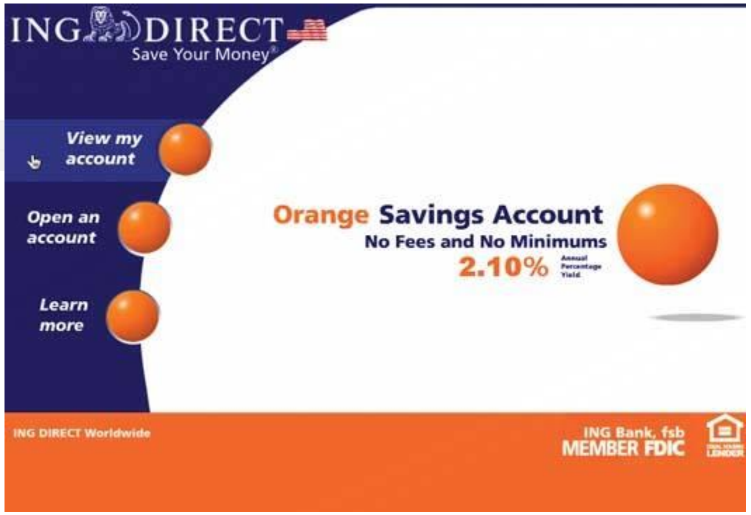
- present only a few task-oriented and descriptive entry points into the UI
- use when the application is task-based or is generally used by first-time/infrequent users
    - don’t use this pattern if the application’s purpose is obvious to everyone and if most users would be irritated by this additional step
- gives the user (especially novice) a clear idea of where to start
    - clearly delineates what the application does
    - make sure the available options match the users’ expectations

clear entry points pattern contd…

- protocol:
  - present entry points as “doors” into the system
    - *unambiguous guides into the application*
    - *help the user develop his/her own sense of context*
  - entry points should collectively cover most of the primary reasons for someone using the application/coming to the website
    - *could be one or two or many depending on application*
    - *label them appropriately for first-time users*
  - present entry points with emphasis proportional to importance
  - splash screens ≠ entry points because they don’t present a decision point to the user

clear entry points examples…





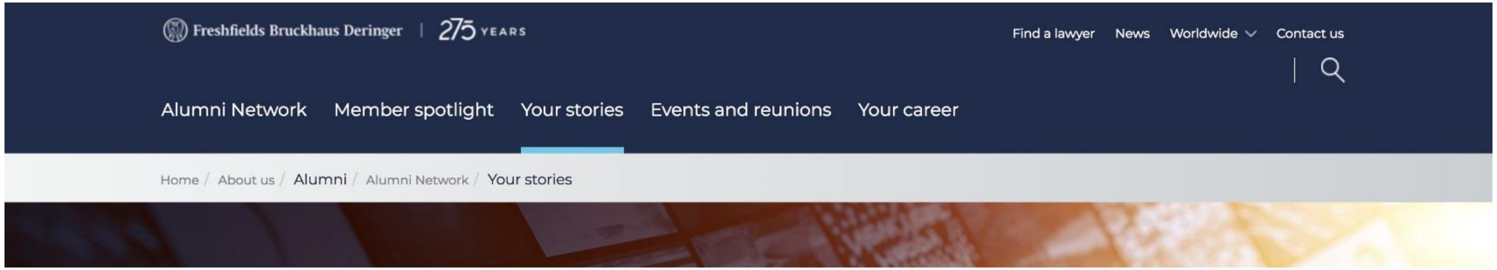
global navigation pattern…

- use a small section of each page to present consistent set of links/buttons that take users to key sections of application
- use when the application has lots of different sections or tools and/or when users are likely to want to move directly from one section to another
    - have to have plenty screen real estate to work with
    - the added elements will not overly clutter the UI
- set of links or buttons that reflect the UI’s highest-order structure makes the structure visible at all times
    - gives overview of the UI and helps users locate what is needed
    - facilitates exploration and easy, one-click transition
    - can add a “you are here” signpost to indicate what section is current

global navigation pattern contd…

- protocol:
  - determine organisational structure
    - *limit sections to what you can sensibly display and label well*
  - global navigation panel should look the same and be located in the same place on every page/window (not dialogue boxes)
  - to show user’s current location, make the current link/button appear different to the rest
  - not every user will use or notice a global navigation panel (so be prepared!)

global navigation examples…

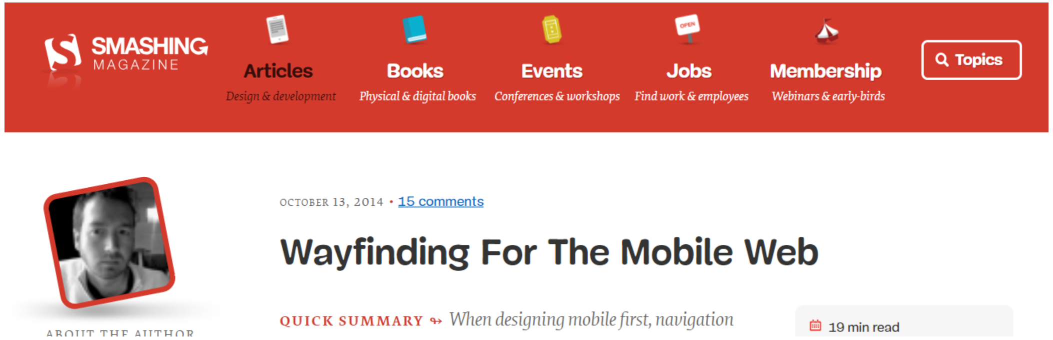


Freshfields Bruckhaus Deringer | 275 YEARS

Find a lawyer News Worldwide Contact us

Alumni Network Member spotlight Your stories Events and reunions Your career

Home / About us / Alumni / Alumni Network / Your stories



SMASHING MAGAZINE

Articles — Design & development
Books — Physical & digital books
Events — Conferences & workshops
Jobs — Find work & employees
Membership — Webinars & early-birds

Topics

ABOUT THE AUTHOR

OCTOBER 13, 2014 • 15 comments

Wayfinding For The Mobile Web

QUICK SUMMARY ↬ When designing mobile first, navigation

19 min read

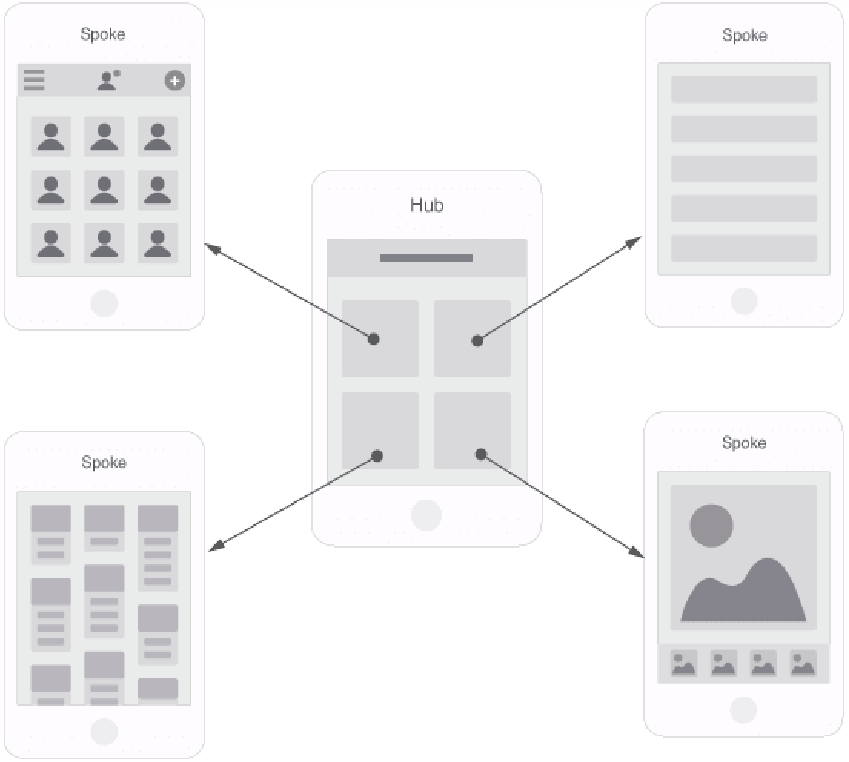
hub and spoke pattern…

- isolates sections of an application into mini-applications, each with one entry point (from the main page) and one way out (to the main page)
- use when the application contains several discrete tasks, sub-applications, etc. all of which can be accessed from one central control page/screen
  - use when you don’t want to connect each of the sub-applications directly to all the others to reinforce the separation, to restrict workflow (and force task completion), to avoid clutter, or due to screen real estate constraints
  - don’t use if you have good reason to allow a user to move directly from one sub-application to another
  - really well suited to small screen devices when used in conjunction with one-window drilldown pattern
- forces user to focus on one thing at a time
  - helps prevent errors (less chance of getting UI into inconsistent state)
  - tighter control over what the UI has to handle (simpler implementation)
  - scales well to handle additional functionality

hub and spoke pattern contd…

- protocol:
  - **spokes**: split content up into self-contained mini-applications (e.g., by task or tool)
  - **hub**: arrange access points/links to the spokes
  - remove all distracting navigational links from the spoke pages/windows and leave only pertinent actions (e.g., exits, help, back, etc.)
  - provide means for user to indicate completion or cancellation at the end of a spoke-based task
    - *these should return the user to the hub*

hub and spoke example…



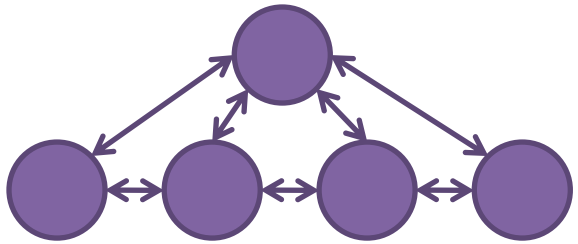
© https://www.smashingmagazine.com/2014/10/wayfinding-for-the-mobile-web/

pyramid pattern…

- links a sequence of pages/windows with Back/Next links
  - often combined with a main page (jumping off point) that links to and from all the pages in the sequence
- use when the application comprises a sequence of pages/windows that a user normally views one after the other
  - e.g., chapters in a book
  - most often paired with one-window drilldown
- reduces number of clicks necessary to get around
  - improves navigation efficiency + highlights sequential relationship among pages
  - most effective if home/main page links included to prevent excessive back/forward button clicks

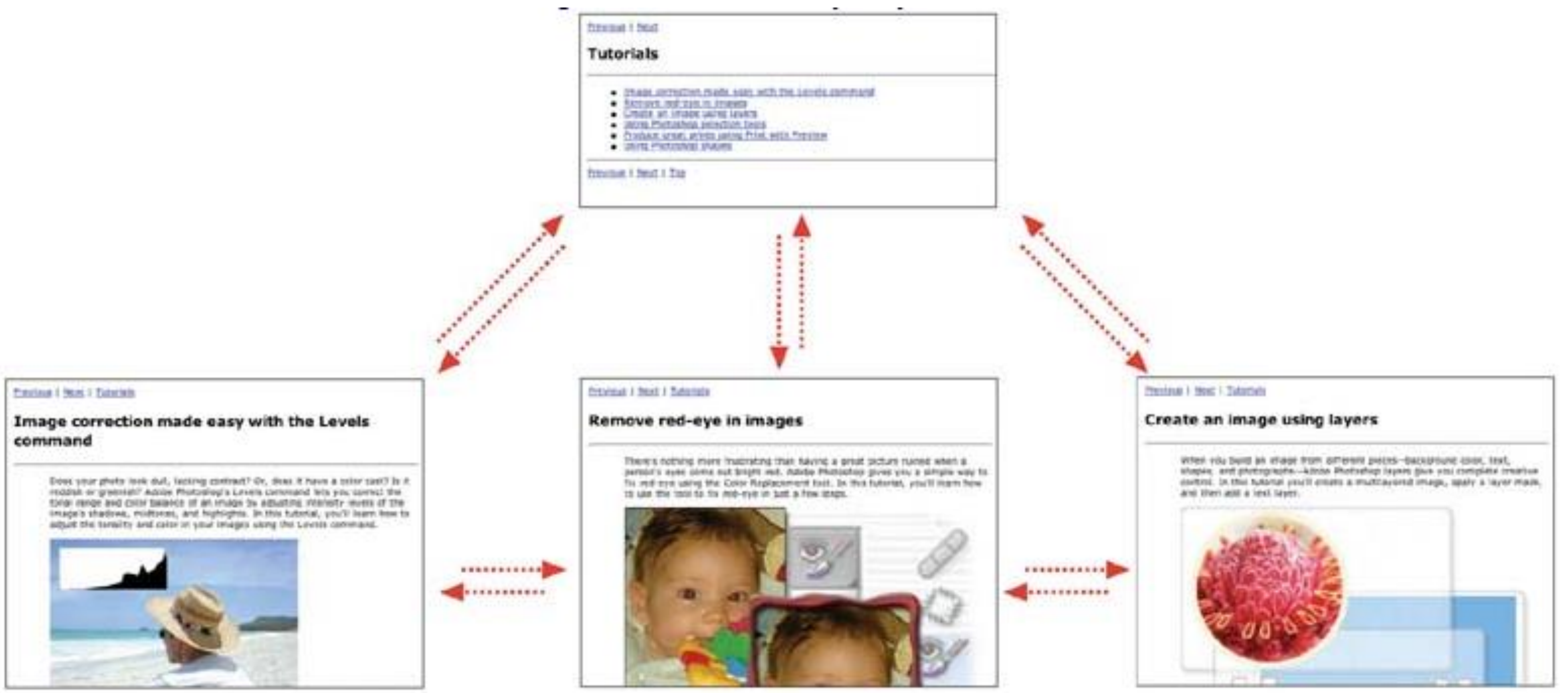
pyramid pattern contd…

- protocol:
  - put *back*, *next*, and *up* buttons or links on each page/window
    - *don‘t have to be labelled as back, next, or up*
  - put links/buttons to each sequence page/window on the main/home page



  - co-locating the links/buttons on screen is useful
    - *minimises mouse motion and establishes spatial memory*
  - if possible, put the next links/buttons in the same place on each page/window
  - can also use this navigational pattern to navigate amongst items within a list

pyramid example…


Previous | Next
Tutorials
Previous | Next | Top
Previous | Next | Tutorials
Image correction made easy with the Levels command
Remove red-eye in images
Create an image using layers

© Tidwell, J., (2005), *Designing Interfaces*, O'Reilly Media Inc.

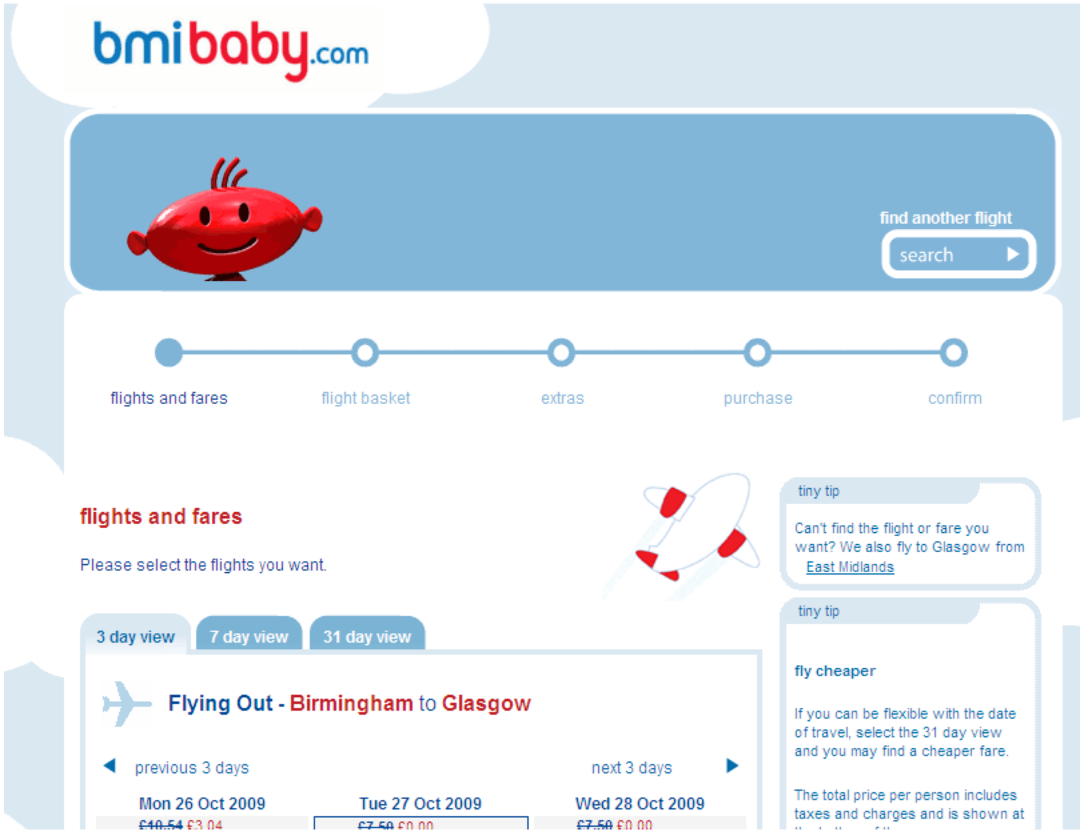
sequence map pattern…

- on each page in a sequence, show a map of all pages in order with a “you are here” indicator
- use for applications with a narrative or an enforced progression (typically linear)
- people do not want to waste time figuring out where they are
- lets people know how much further they have to go in a progression
    - gives indication of the past-present-future of activity

sequence map pattern contd…

- protocol:
  - place small map of pages in the sequence somewhere near the edge of the page/window (often along the top)
    - *keep it to one line or column if possible*
    - *clearly highlight the current location indicator*
  - use labels and/or step numbers to identify the pages/windows in the sequence map

sequence map example…



bmibaby.com

find another flight

search

flights and fares · flight basket · extras · purchase · confirm

flights and fares

Please select the flights you want.

3 day view | 7 day view | 31 day view

Flying Out - Birmingham to Glasgow

previous 3 days   next 3 days

Mon 26 Oct 2009	Tue 27 Oct 2009	Wed 28 Oct 2009
~~£10.54~~ £3.04	~~£7.50~~ £0.00	~~£7.50~~ £0.00

tiny tip

Can't find the flight or fare you want? We also fly to Glasgow from East Midlands

tiny tip

fly cheaper

If you can be flexible with the date of travel, select the 31 day view and you may find a cheaper fare.

The total price per person includes taxes and charges and is shown at

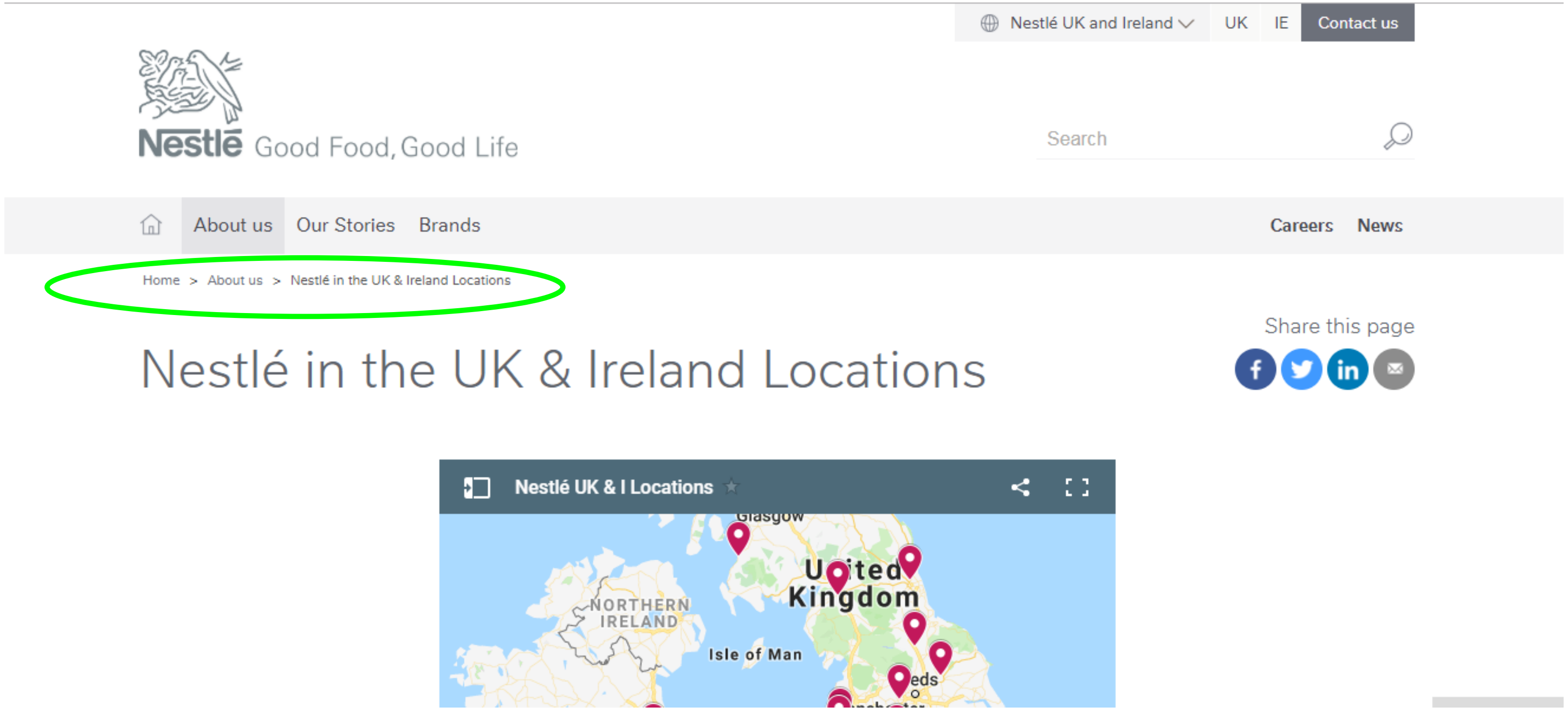
breadcrumbs pattern…

- on each page (in a hierarchical structure) show a map of all the parent pages, up to the main page
- use for applications with a basic tree or hierarchical structure
  - best if there aren’t too many links between tree elements
  - frequently used in applications with one-window drilldown structure
  - alternative to sequence maps where the sequence map would become too unwieldy
- show a path from root (main page) to current position via all the levels in the hierarchy
  - helps users determine where they are in the structure (especially if they have jumped several levels in one go due to a given course of action)
- breadcrumbs are about context not history
  - gives indication of the current location relative to overall structure NOT the path taken to get there
- since clickable, they are navigation tool in own right

breadcrumbs pattern contd…

- protocol:
    - near edge (usually top but can also be at the bottom) of the page, put a line of text or icons indicating the current level in the structural hierarchy reading from left to right
        - *between the icons put a graphic (e.g., an arrow) to indicate movement from one level to the next*
    - labels for each level indicator should be titles of corresponding page/window
        - *users should recognise them if they have already visited them*
        - *users should be able to guess at their content based on the label even if they are unfamiliar with them*

breadcrumbs example…



Nestlé UK and Ireland UK IE Contact us

Nestlé Good Food, Good Life

Search

About us Our Stories Brands Careers News

Home > About us > Nestlé in the UK & Ireland Locations

Share this page

Nestlé in the UK & Ireland Locations

page layout…

- the art of manipulating the users' attention on a page to convey meaning, sequence, and points of interaction
  - done to help users extract meaning
- 5 major elements of page layout:
  - **visual hierarchy** » most important content should stand out the most and vice versa & user should be able to work out the informational structure of the page from its layout (see Design Principles lectures)
  - **visual flow** » considers the track of users' gaze as they scan a page of information (see Design Principles lectures)
    - *well designed visual hierarchy establishes focal points on the page*
    - *visual flow leads the users' eyes from one focal point to another*
    - *should be able to design for visual flow such that you ensure people follow the right sequence – i.e., designer should set up the right sequence of focal points*
  - **grouping and alignment** (see Design Principles lectures)
  - **how to put the above three elements together**
  - **how to use dynamic displays**

page layout patterns…

- patterns that consider the visual hierarchy of the whole page:
  - visual framework
    - *consider early during design as it affects all the major windows/pages in a UI*
  - centre stage
- patterns that represent ways of chunking content:
  - tiled sections
  - card stack
  - closable panels*
  - movable panels*
- patterns that draw on concepts of visual flow, alignment, etc:
  - right/left alignment*
  - diagonal balance*

*** read Tidwell, J., (2005), Designing Interfaces, O’Reilly Media Inc. Chapter 4**

page layout patterns contd…

- miscellaneous:
  - property sheet*
    - *as much about content and interaction as it is about layout BUT when knowledgeable users recognise that a page has a property sheet, their expectations are strongly set*
- patterns that deal with dynamic aspects of content layout:
  - responsive disclosure*
  - responsive enabling*

  $\left.\begin{array}{l}\text{responsive disclosure*}\\ \text{responsive enabling*}\end{array}\right\}$ reveal/enable items in response to user actions/input
  - liquid layout* - resize window contents as user resizes window

*** read Tidwell, J., (2005), Designing Interfaces, O'Reilly Media Inc. Chapter 4**

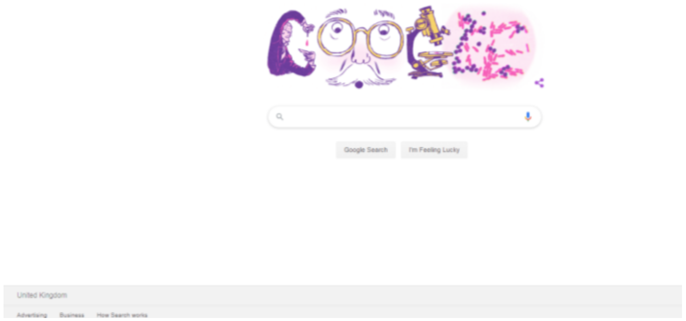
visual framework pattern…

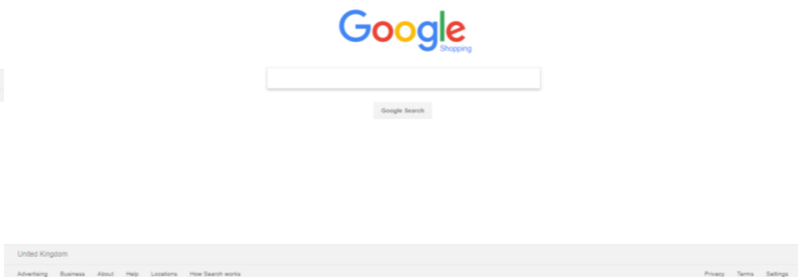
- design each page to use the same basic layout, colours, and stylistic aspects
  - give the design enough flexibility to handle varying page content
- use when the application has multiple windows/pages and you want it to appear as a cohesive unit
- gives the user a sense of familiarity when UI pages use consistent colour, font, and layout, and where signposts are in consistent locations
  - helps users know where they are and where to find things
  - reduces cost of context switch between pages
  - strong visual framework helps content stand out more – constant aspects are habituated

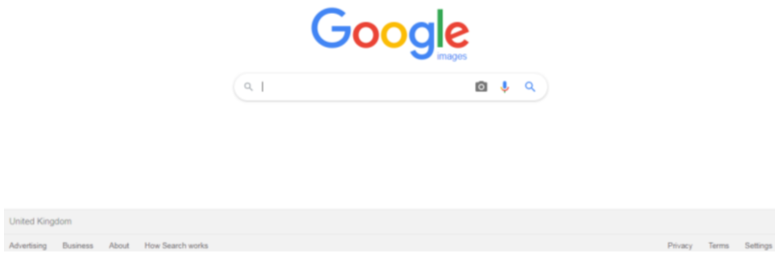
visual framework pattern contd…

- protocol:
  - decide on an overall “look-and-feel” for all pages
    - *home pages or main windows can be different but should share some characteristics with the rest of the application*
    - *consider: colour, fonts, writing style and grammar*
  - where applicable, all pages should share:
    - *signposts*
    - *navigational devices and navigational patterns*
    - *spacing and alignment*
    - *overall layout (grid)*
      - layout grid = structural template for the layout of a set of pages

visual framework example…







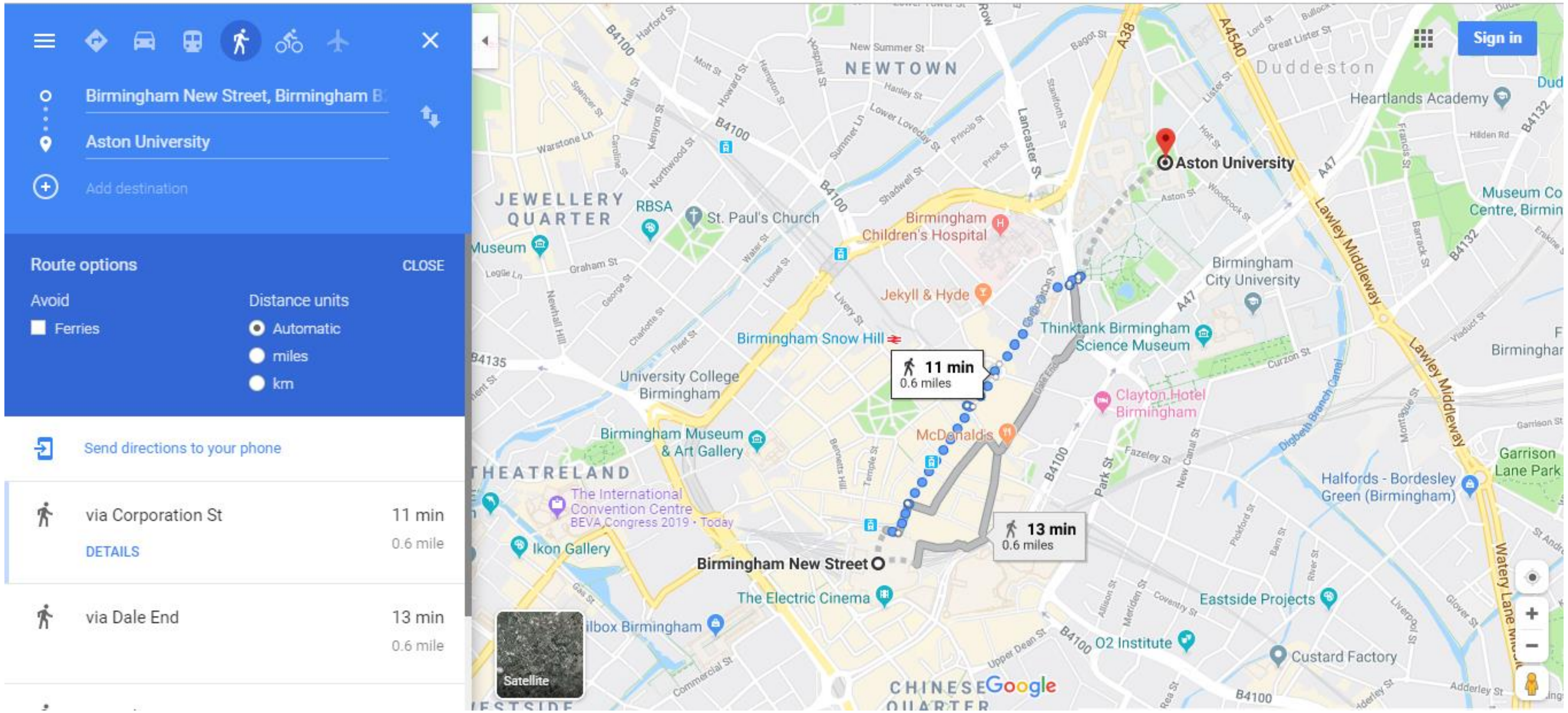
centre stage pattern…

- put the most important part of the UI into the largest part of the page/window
    - group secondary content around it in smaller panels
- use when the application has to show coherent information to the user to let him/her edit an object or perform a given task
    - e.g., graphics editors
- focuses the users' attention on the most important information (task)
    - unambiguous central object focuses users' attention
    - centre stage establishes the purpose of the page/UI
    - users will assess the peripheral items in terms of how they relate to the aspect which has centre stage

centre stage pattern contd…

- protocol:
  - set up a visual hierarchy with the centre stage dominating all else
  - consider:
    - **size**: centre stage should be at least double the width of the peripheral items in the left/right margins and double the height of its top and bottom margins (although this could be resized by the user)
    - **colour**: centre stage colour should contrast with the peripheral information/objects – white against grey often works well
    - **headlines**: big headlines = focal points which can draw users' attention to the top of the centre stage
    - **context**: fit with users' preconceptions in terms of what the users expect to see in centre stage
  - ironically, doesn't matter where you put the "centre" stage – can work in top, bottom, left, or right positions relative to the page as a whole – as it should dominate wherever it is placed on the page

centre stage example…



Birmingham New Street, Birmingham B

Aston University

Add destination

Route options CLOSE

Avoid

- [ ] Ferries

Distance units

- [✓ Automatic]
- miles
- km

Send directions to your phone

via Corporation St	11 min	0.6 mile
DETAILS		
via Dale End	13 min	0.6 mile

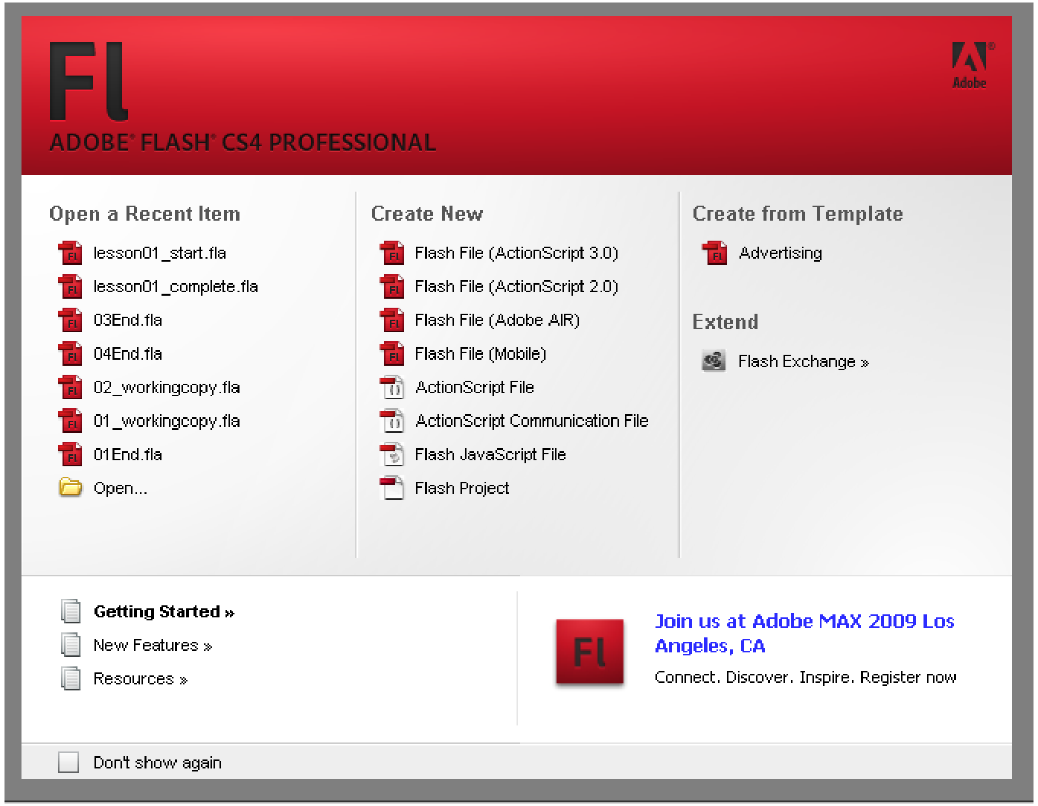
tiled sections pattern…

- identify separate sections of content by giving each a visually strong title and then laying them out together on the same page
- use when there is a lot of content to display on the page but you want to ensure that it is easy to scan the page and to find information
    - can use thematic or task-based grouping of content
- well thought out and labelled sections present the content in easy-to-manage chunks
    - makes information architecture obvious
    - guides user’s eye along the page content

tiled sections pattern contd…

- protocol:
    - carefully consider the information architecture – split content up into coherent chunks with clear names
        - *if you are struggling to find clear names, suggests that the content grouping in a section does not represent a meaningful chunk of the overall content*
    - present chunk titles in font that stands out from the rest of the content
    - use whitespace to visually separate chunks/sections
    - carefully use boxes etc. to add visual separation

tiled sections example…



Open a Recent Item

- lesson01_start.fla
- lesson01_complete.fla
- 03End.fla
- 04End.fla
- 02_workingcopy.fla
- 01_workingcopy.fla
- 01End.fla
- Open...

Create New

- Flash File (ActionScript 3.0)
- Flash File (ActionScript 2.0)
- Flash File (Adobe AIR)
- Flash File (Mobile)
- ActionScript File
- ActionScript Communication File
- Flash JavaScript File
- Flash Project

Create from Template

- Advertising

Extend

- Flash Exchange »

Getting Started »
New Features »
Resources »

Join us at Adobe MAX 2009 Los Angeles, CA
Connect. Discover. Inspire. Register now

Don't show again

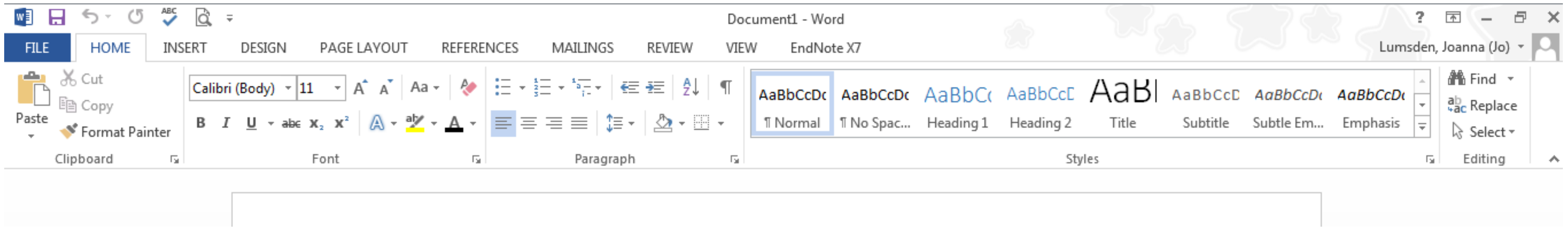
card stack pattern…

- allocate sections of content to separate panels or cards and stack them up so only one is visible at a time
  - use tabs to give users access to the hidden panels/cards
- use when there is too much material to present at the same time on a single page and where the users’ attention would become distracted by the extent of controls/text across a UI
  - often used when tiled sections would end up being too big to fit onto the page at the same time
- the labelled card structure separates content into easily digestible chunks
  - users are very familiar with tabs – the most common form of card stack

card stack pattern contd…

- protocol:
  - carefully consider the information architecture – split content up into coherent chunks with clear names
    - *if you get the split wrong, users will have to switch back and forth between different cards!*
  - choose a presentation:
    - *tabs are good for 6 or less cards – don't use double rows of tabs as these are never easy to use – and scroll horizontally if you can't fit them all into one row at once*
    - *vertical tabs allow you to utilise a tall space that can't accommodate normal tab pages*
    - *a left-hand column of names is also an option – can fit a lot of cards into a single column and allows you to organise cards into a hierarchy which is not possible with standard tabs*
      - becomes like the two-panel selector
    - *drop-down list takes up less space but at the expense of clarity and users might not recognise it as a navigational device as this is not its normal usage*

card stack example…



action and command patterns…

- patterns for presenting actions:
  - button groups
  - action panel
  - prominent “done” button
  - smart menu items*
- patterns for non-instantaneous actions:
  - preview*
  - progress indicator*
  - cancelability*
- patterns for sequences of actions (not easy to implement):
  - multi-level undo*
  - command history*
  - macros*

*** read Tidwell, J., (2005), Designing Interfaces, O’Reilly Media Inc. Chapter 5**

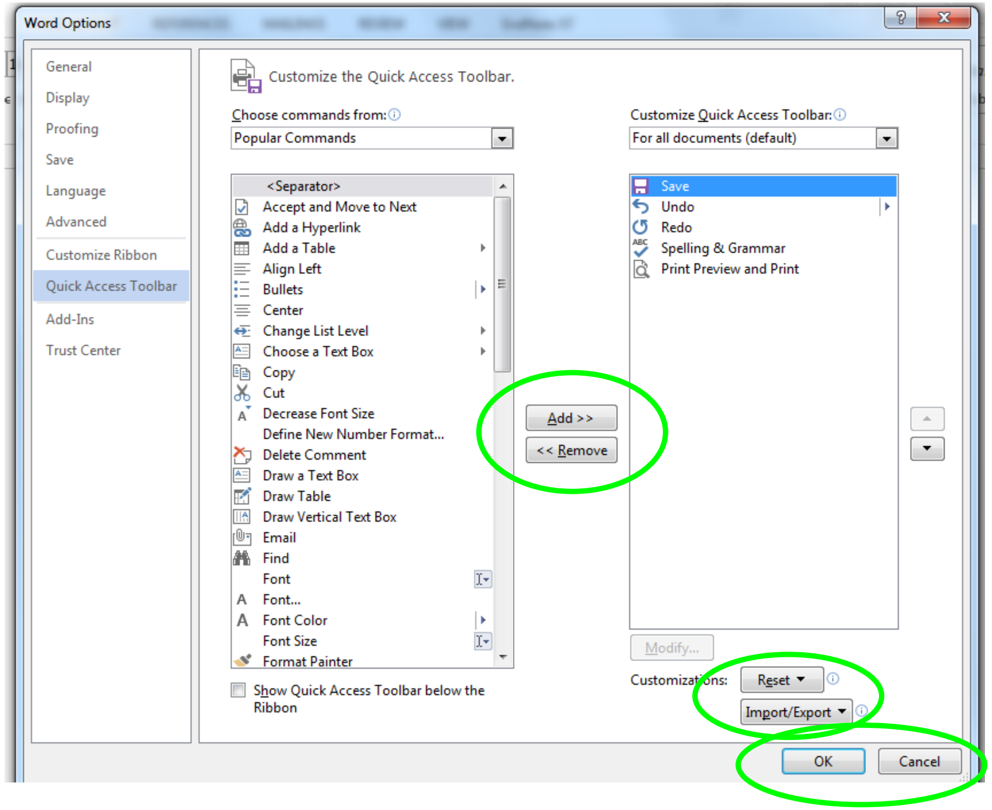
button groups pattern…

- present related actions as clusters of aligned buttons
- use when you present 2 – 5 related actions
    - e.g., OK, Cancel, Apply, Close
    - e.g., Move Up, Move Down, Delete
- help make interface self explanatory
    - well-defined clusters are easy to identify
    - Gestalt principles apply here
        - *proximity & visual similarity = relatedness*
        - *size and alignment = create large composite object = closure*

button groups pattern contd…

- protocol:
  - all buttons in group should be same width and height (unless label lengths vary considerably)
    - *try and stick to single columns or rows of buttons*
  - if buttons act on same object put the group to the right of the object
    - *users have "blind spot" at the bottom of complex UI elements*
  - if buttons apply to whole page/dialogue box – e.g., Close or OK – follow the style guide for the platform where users will be habituated to look for them

button groups example…



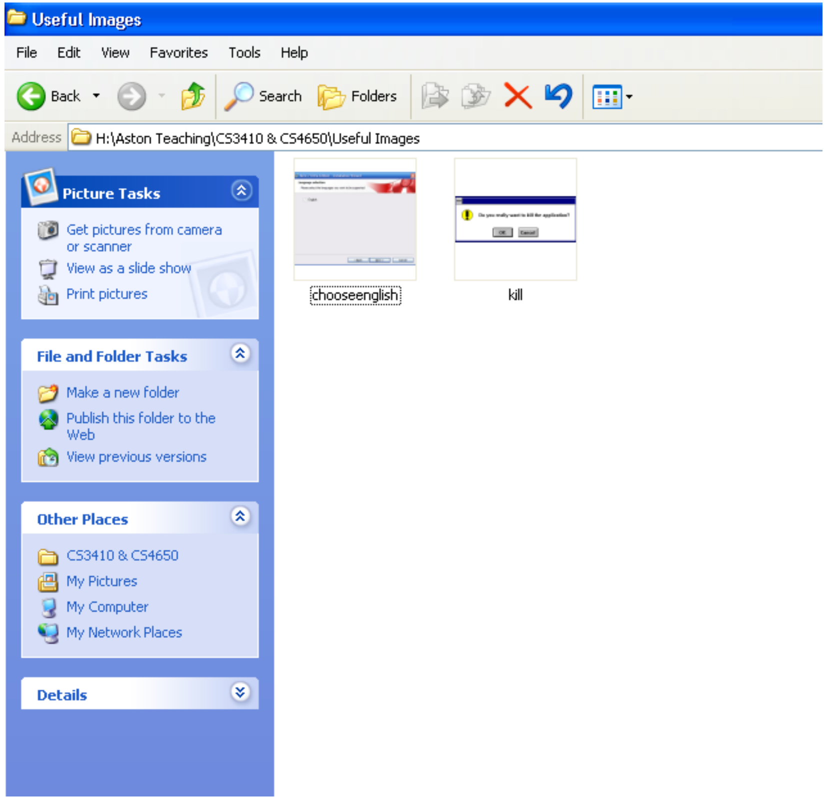
action panel pattern…

- instead of using menus, present large group of related actions on a UI panel that is always visible
- use when you have too many actions to use a button group
    - could put them in a menu but perhaps you don't have a menu bar in the UI or you simply want to make the actions more obvious
    - alternatively, the set of possible actions may be too complex even for a menu – cascading menus can be hard for users to manipulate
    - takes up a lot of screen real estate so not good for small devices
- ensures actions are visible and gives designer freedom of presentation
    - action panels are really just permanently displayed menus
    - makes interface functionality discoverable
    - familiarity via web page design

action panel pattern contd…

- protocol:
  - placing the panel on the UI
    - *place below or to the side of the target of the action (remember law of proximity)*
    - *if panel is closable, make it easy to reopen*
    - *let it be dynamic to reflect current state of the application*
  - structuring the actions – possible options:
    - *simple lists, multi-column lists, categorised lists (probably task-centred), tables or grids, closable panels, or any combinations of the above*
    - *present groups linearly – think about accessibility for screen reader*
  - labelling the actions
    - *use text, icons, or a combination of both to label actions*
    - *text labels can be longer here than a traditional toolbar or menu would allow*
    - *actions don't have to look like buttons (even if that is how they are implemented)*

action panel example…



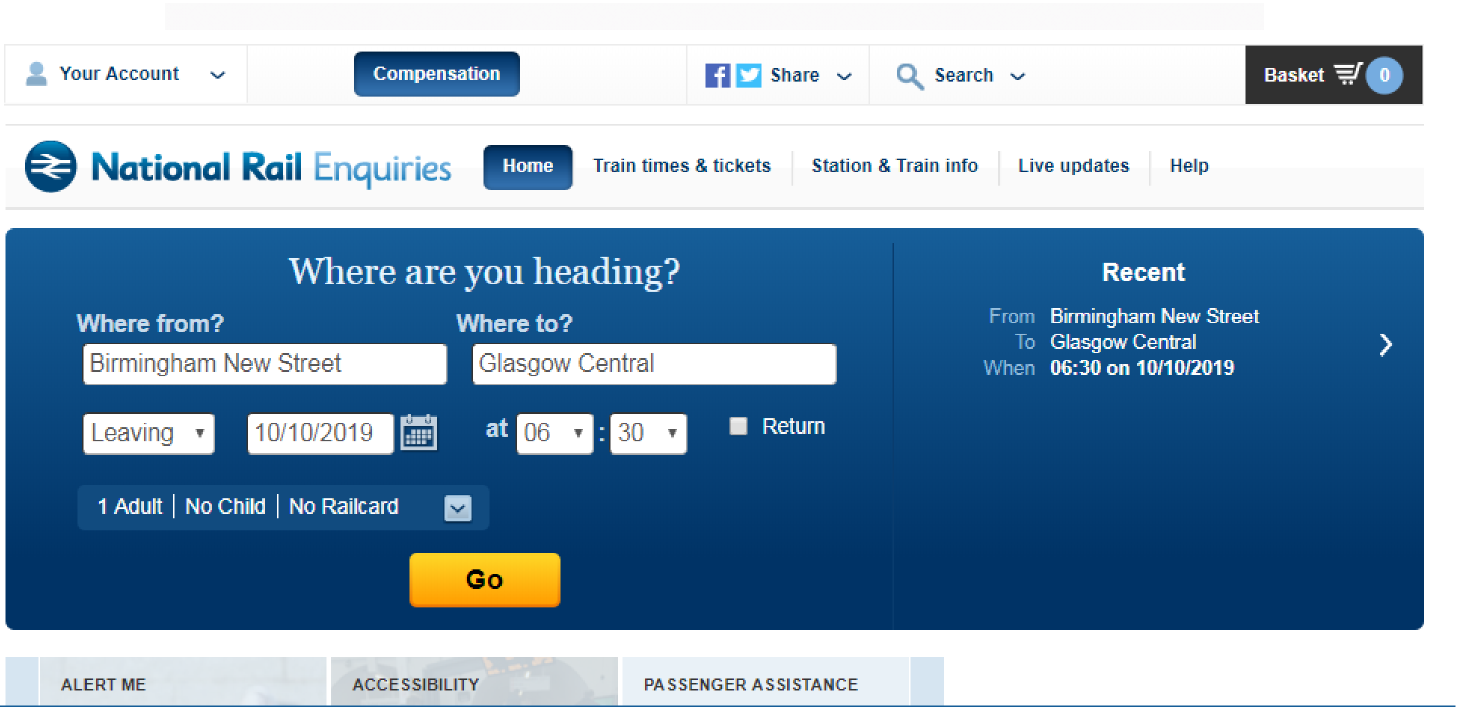
prominent ‘done’ button pattern…

- place button that concludes a transaction at the end of the visual flow and make it prominent
- use when you need a button like “done”, “submit” or “OK”
    - in general, anywhere where a button is required to conclude a transaction sequence
- well understood and obvious
    - sense of closure
    - draws on concepts of visual hierarchy, visual flow, grouping, and alignment to get this right

prominent ‘done’ button pattern contd…

- protocol:
  - make button look like a button not a link
    - *make sure it stands out on the page*
  - better to use textual labels than icons for this button
  - place the button where the user will most likely find it
    - *trace task flow to place the button just after the last step*
    - *usually at bottom right but observe standards*
  - make sure button is close to the last field or control on the page – if it is too far away users may not find it immediately upon finishing their work

prominent ‘done’ button example…



Where from? Birmingham New Street
Where to? Glasgow Central
Leaving 10/10/2019 at 06 : 30 Return
1 Adult | No Child | No Railcard

Recent
From Birmingham New Street
To Glasgow Central
When 06:30 on 10/10/2019

key points…

- essential to get the structure of an application and the content and navigation within that application right in order to maximise usability and effectiveness
- variety of application structure patterns to guide the physical structure of the UI design
- navigation is critical to use of an application and comes with associated costs
- variety of navigation/orientation patterns to guide the navigational elements of a UI design
- layout of page elements can be guided by design principles (lectures 7 & 8) and related page layout patterns

Individual Reading:

Tidwell, J., (2005), Designing Interfaces, O’Reilly Media Inc. Chapters 2 - 3