

# User Interface Design

# What is User Interface

- Basically a page or screen in a software application
- A part of software that receives input
  - From the users
- Must have the facility to get input
  - Through keyboard
  - Mouse click
  - Voice
- What kind of Interface?
  - Interface for processing part of the software system
  - Sends data to the to the data processing module

# Why is UI important

- The users of the system operates through this interface
- The performance of the system depends on the user interface
- Satisfaction of the user interface is important
- It is important to design UI first
- This is a sketch of what user wants
- Easy to convey the users what they will get

# Usability in UI

- The effectiveness with which users can accomplish tasks in a (software) system, as measured by
  - Learnability: is it easy to learn?
  - Efficiency: once learned, is it fast to use?
  - Safety: are errors few and recoverable?

# Relative importance of usability dimensions

- Depends on the user
  - Novices need learnability.
  - Experts need efficiency.
  - But no user is uniformly a novice or an expert.
- Depends on the task
  - Missile launchers need safety.
  - Subway turnstiles need efficiency.



# Usability matters: the cost of getting it wrong

*50% of all “malfunctioning” electronic devices returned to stores are in full working order, but users can't figure out how to operate them.*

*[Elke den Ouden, 2006]*



**Three Mile Island:**  
nuclear reactor  
meltdown caused by an  
ambiguous user  
interface

# Designing UI: A good user interface is hard to design ...

- You are not the user
  - Most software engineering is about communicating with other programmers.
  - UI is about communicating with users.
- Users are always right ...
  - Consistent problems are the system's fault.
- Except when they aren't
  - Users don't always know what they want.

# Achieving usability: best practices

- User testing and field studies
- Evaluations and reviews by UI experts
- Prototyping
  - Cheap, throw-away implementations
  - Low-fidelity: paper prototypes
  - Medium-fidelity: code prototypes

**Key to success:** good UI focuses on the user, not the developer or the system.

# What is prototyping? Why do it?

- **Prototyping:** creating a scaled-down or incomplete version of a system to demonstrate or test its aspects.
- **Benefits** of prototyping:
  - aids UI design
  - help discover requirements
  - help discover test cases and provide a basis for testing
  - allows interaction with user to ensure satisfaction
  - team-building

# Some prototyping methods

- Code prototyping
  - implement a "quick" / incomplete version of a UI
- Prototyping with UI builders (e.g., Visual Studio)
  - draw a GUI by dragging/dropping UI controls on screen
- **Paper prototyping**
  - a paper version of a UI

# Why paper prototyping?

- Much faster to create and change than code
- More visual bandwidth (can see more at once)
- More conducive to working in teams
- Can be done by non-technical people
- Feels less permanent or final



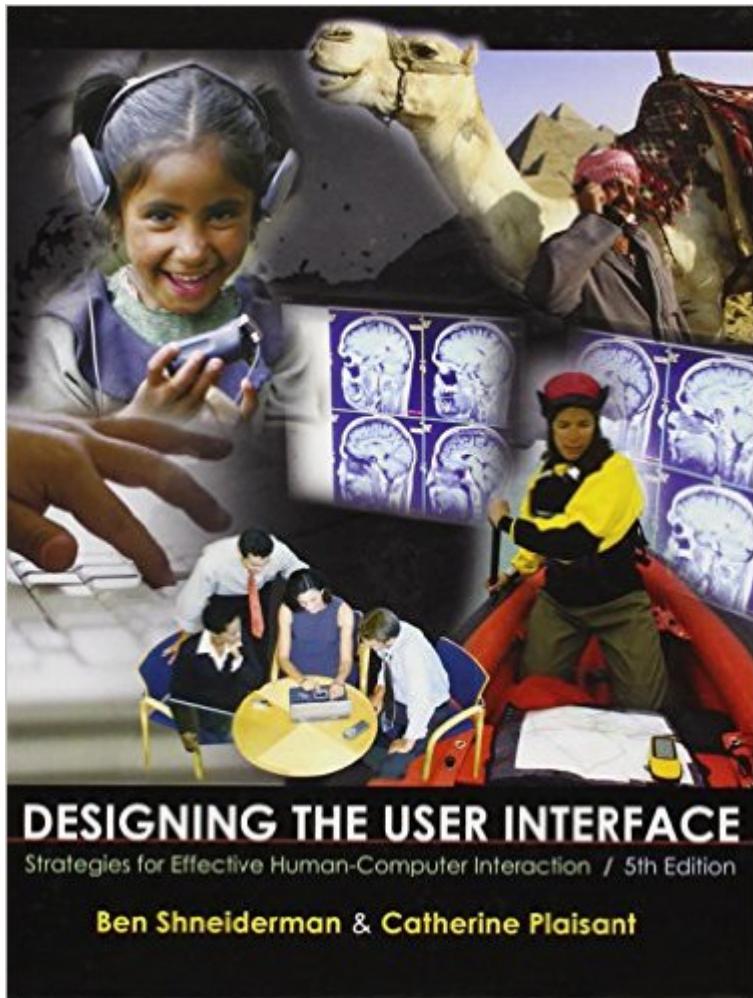
# When to do prototyping?

- During or after requirements but before design
  - helps uncover requirements and upcoming design issues
  - shows us **what** is in the UI, but also shows us details of **how** the user can achieve goals in the UI



# UI Design Considerations

# Schneiderman's 8 Golden Rules



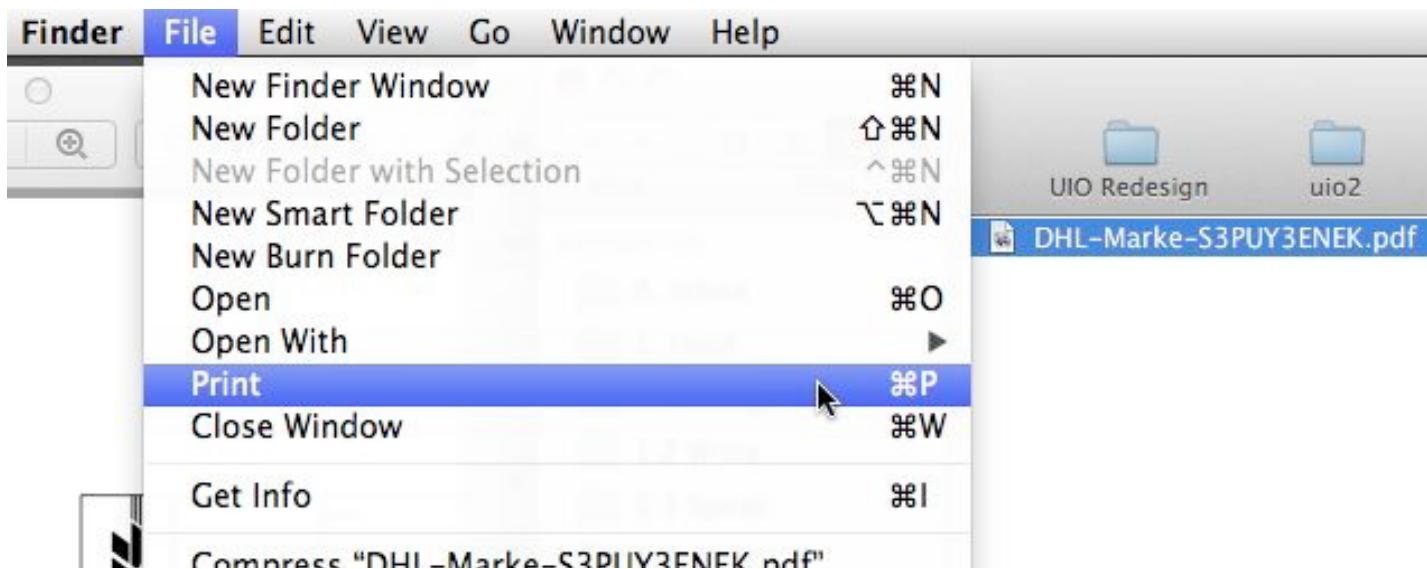
# Rule 1: Strive for consistency.

*Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.*



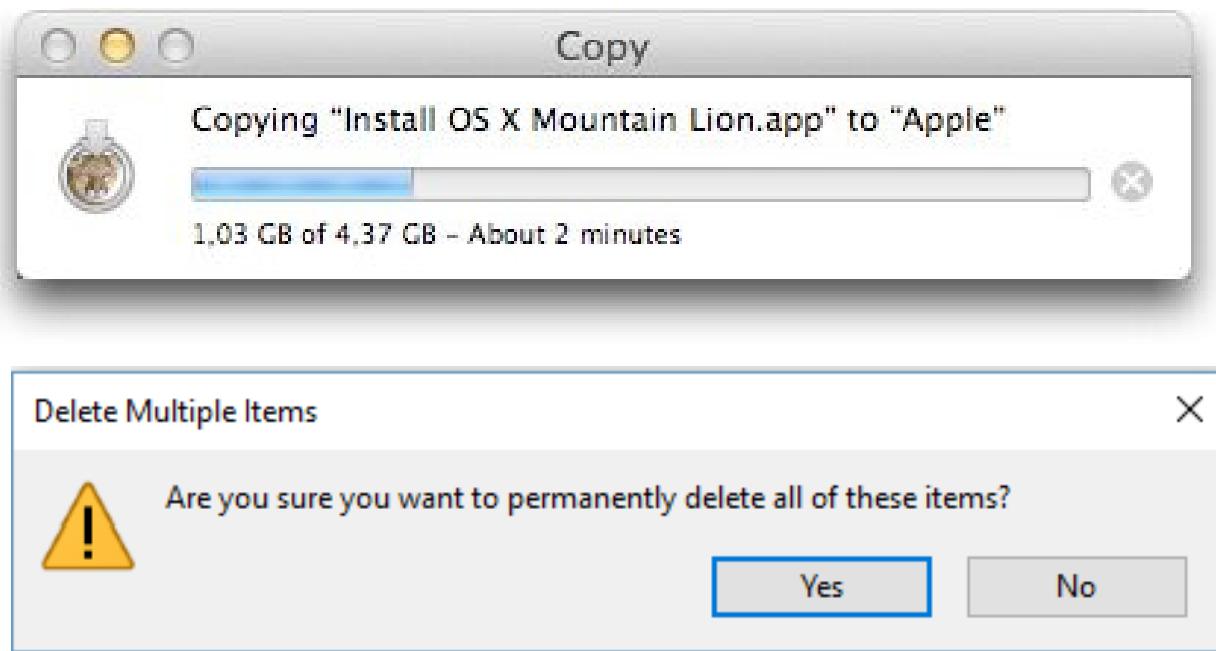
## Rule2: Enable frequent users to use shortcuts

*As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.*



# Rule 3: Offer informative feedback

*For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.*



# Rule 4: Design dialog to yield closure

*Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.*

The screenshot shows a sequence of three dialog boxes from an Amazon checkout process:

- Step 1: Choose a shipping address**

Your addresses

  - Amiangshu Bosu (selected)
  - Add a new address

Use this address
- Step 2: Payment method**

VISA Visa ending in

Billing address: Same as shipping address. Change

Add a gift card or promotion code

Enter code Apply
- Step 3: Items and shipping**

Estimated delivery: Mar. 15, 2016

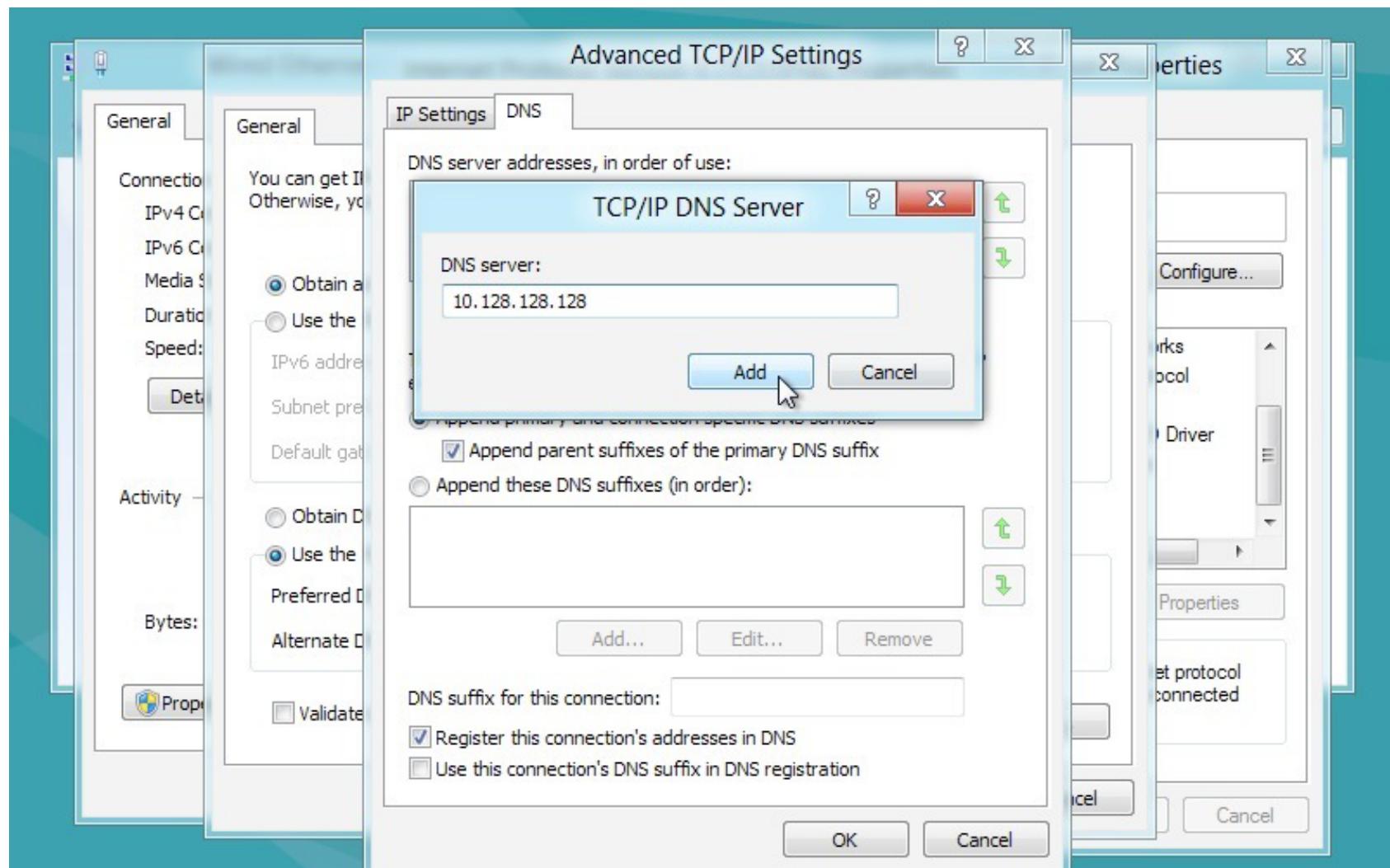
View

Amazon Echo

Informative feedback and closure elements include:

- A yellow "Use this address" button in Step 1.
- A yellow "Use this address" button in Step 2.
- A "Continue to step 3 to finish checking out. You'll have a chance to review and edit your order before it's final." message in Step 2.
- An "Order Summary" section in Step 2 listing items, shipping & handling, total before tax, estimated tax, and order total.
- A "By placing your order, you agree to all of the terms found here." link and a "How are shipping costs calculated?" link in Step 2.

# Rule 4: Design dialog to yield closure



# Rule 5: Offer simple error handling

*As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.*

Your Name

Andy Geschäftsleitung

Edit

Apple ID and Primary Email Address

abc@something.com

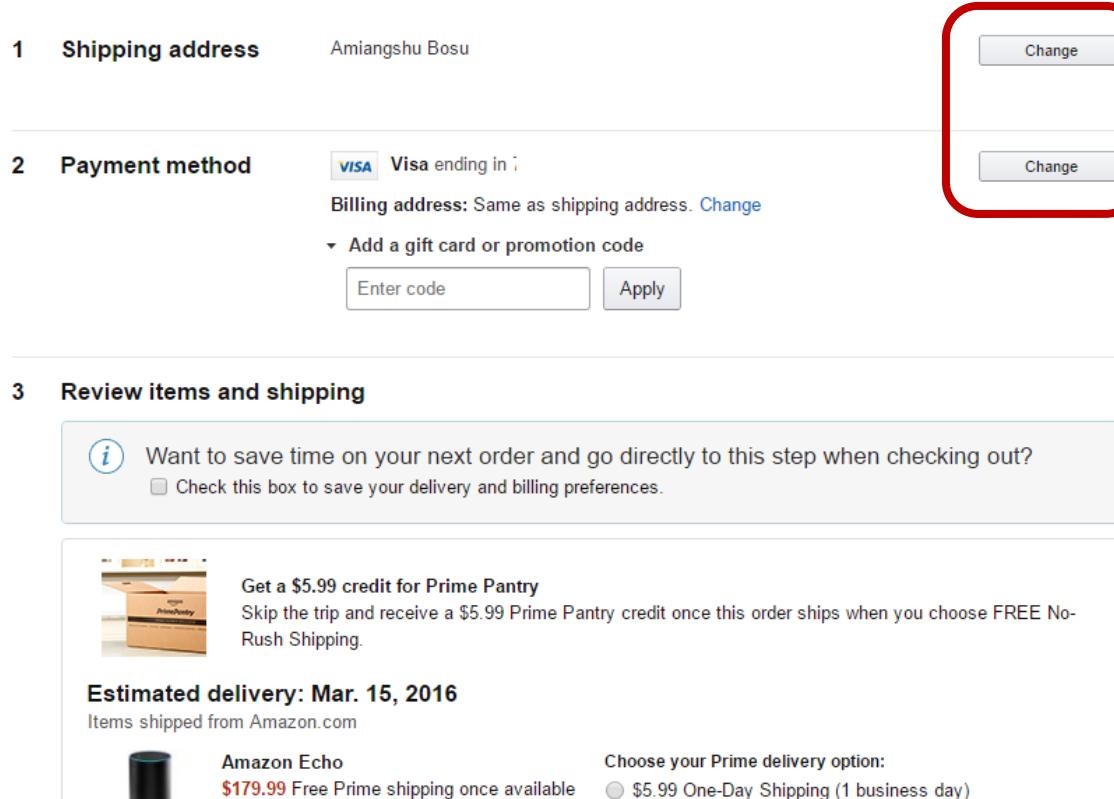
You'll use your new Apple ID to sign in to services such as iTunes, the Online Store, and Game Center. Please note that you might be required to verify your email address before you can start using your new Apple ID.

Apple ID must:

- Be a valid email address
- Not already be in use
- Not use a domain owned by Apple

# Rule 6: Permit easy reversal of actions

*This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.*



The screenshot shows a step-by-step checkout process:

- 1 Shipping address:** Shows "Amiangshu Bosu". To the right is a "Change" button.
- 2 Payment method:** Shows a "Visa" logo and "Visa ending in 1". Below it says "Billing address: Same as shipping address. [Change](#)". There's also a dropdown for "Add a gift card or promotion code" with an "Enter code" input field and an "Apply" button. Both the "Change" link and the "Change" button are highlighted with a red box.
- 3 Review items and shipping:** This section includes:
  - A note: "Want to save time on your next order and go directly to this step when checking out?  Check this box to save your delivery and billing preferences."
  - An offer: "Get a \$5.99 credit for Prime Pantry. Skip the trip and receive a \$5.99 Prime Pantry credit once this order ships when you choose FREE No-Rush Shipping." It features an image of a Prime Pantry box.
  - Delivery information: "Estimated delivery: Mar. 15, 2016" and "Items shipped from Amazon.com".
  - A product item: "Amazon Echo" at "\$179.99" with "Free Prime shipping once available".
  - Delivery options: "Choose your Prime delivery option:" with a selected radio button for "\$5.99 One-Day Shipping (1 business day)".

# Rule 7: Let the user be in control.

*Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.*



# Rule 8: Reduce short-term memory load on the user.

*The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.*

The screenshot shows the Amazon checkout interface with three main steps:

- 1 Shipping address**: Shows the shipping address "Amiangshu Bosu".
- 2 Payment method**: Shows a VISA card ending in 1234. Billing address is same as shipping. Options to add a gift card or promotion code.
- 3 Review items and shipping**: Shows an item: "Amazon Echo" (\$179.99). Estimated delivery is Mar. 15, 2016. Order total is \$191.24.

**Checkout (1 item)** (highlighted in red box)

**Place your order**

By placing your order, you agree to Amazon.com's [privacy notice](#) and [conditions of use](#).

**Order Summary**

Items:	\$179.99
Shipping & handling:	\$0.00
Total before tax:	\$179.99
Estimated tax to be collected:	\$11.25
<b>Order total:</b>	<b>\$191.24</b>

By placing your order, you agree to all of the terms found [here](#).  
How are shipping costs calculated?  
Prime shipping benefits have been applied to your order.

**Estimated delivery: Mar. 15, 2016**  
Items shipped from Amazon.com

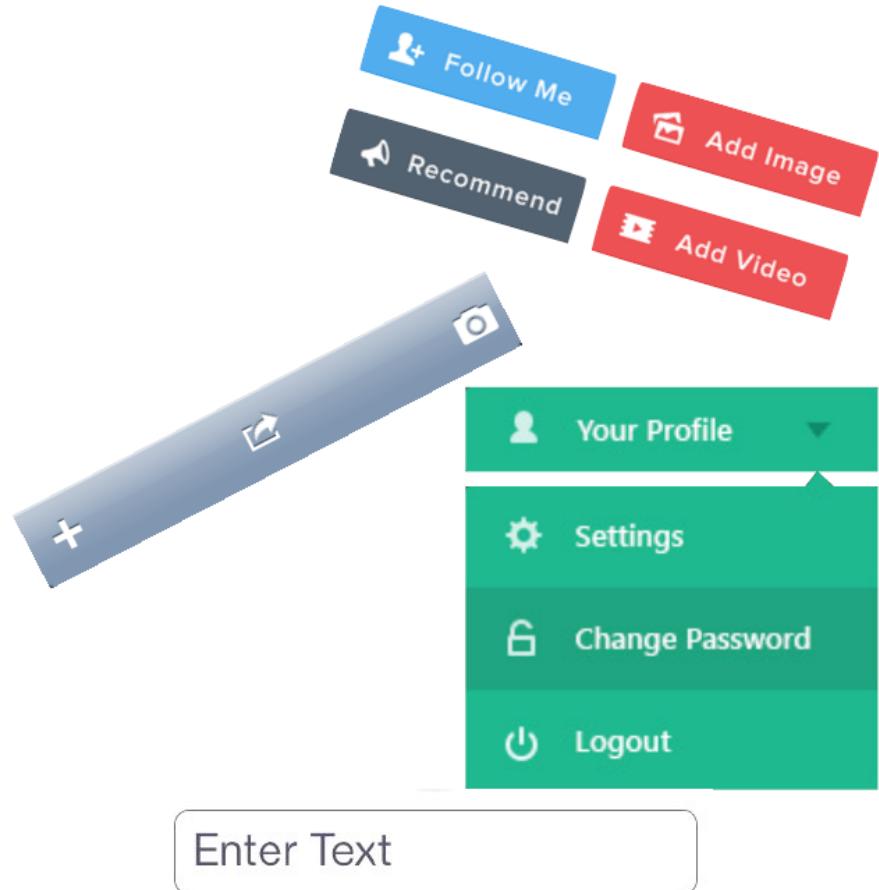
**Amazon Echo**  
\$179.99 Free Prime shipping once available  
Qty: 1 Add a gift receipt  
Sold by: Amazon Digital Services LLC  
In stock on March 10, 2016.  
Choose your Prime delivery option:  
 \$5.99 One-Day Shipping (1 business day)  
 FREE Two-Day Shipping (2 business days)  
 FREE Standard Shipping (4-5 business days)  
 FREE No-Rush Shipping (5 business days)  
Get a \$5.99 credit for Prime Pantry. [Details](#)

and see other gift options  
 Pay in 5 monthly payments of \$36.00

# UI Design components

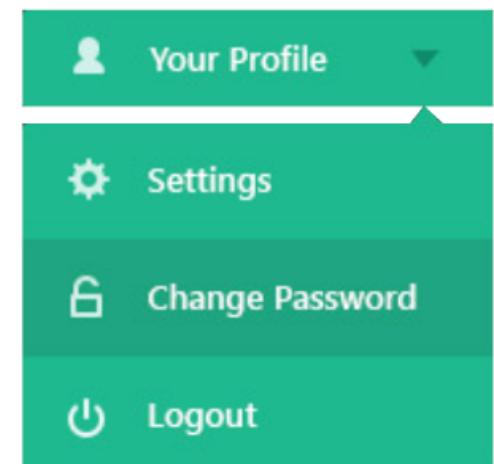
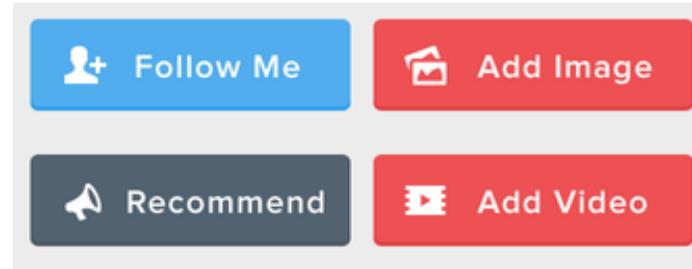
# When to use?

- a button?
- a check box?
- a radio button?
- a text field?
- a list?
- a combo box?
- a menu?
- a dialog box?



# UI design: buttons, toolbars, menus

- Use **buttons** for single independent actions that are relevant to the current screen.
  - Use button text with verb phrases such as "Save" or "Cancel", not generic: "OK", "Yes", "No"
  - Use Mnemonics or Accelerators (Ctrl-S)
- Use **toolbars** for common actions.
- Use **menus** for infrequent actions that may be applicable to many or all screens.
  - Users hate menus! Try not to rely too much on menus. Provide another way to access the same functionality (toolbar, hotkey, etc.)



# UI design: check boxes and radio buttons

- Use **check boxes** for independent on/off switches.
- Use **radio buttons** for related choices, when only one choice can be activated at a time.

1. Do you have pets?

Yes

No

2. Which pets do you have?

Dog

Cat

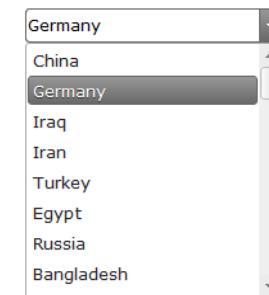
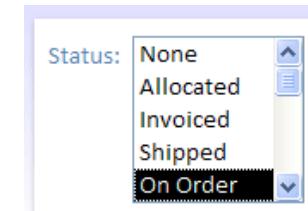
Lizard

Bird

# UI design: text fields, lists, combo boxes, sliders

- Use **text fields** (usually with a label) when the user may type in anything they want.
- Use **lists** when there are many fixed choices (too many for radio buttons); all choices visible on screen at once.
- Use **combo boxes** when there are many fixed choices; don't take up screen real estate by showing them all at once.
- Use a **slider** or **spinner** for a numeric value.

Email or Phone      Password



# UI design: dialogs and panes

- Use a **tabbed pane** when there are many screens that the user may want to switch between at any moment
- Use **dialog boxes** or **option panes** to present temporary screens or options
  - “modal” dialog box prevents any other action

