Display of Information

- All humans have more or less the same visual processing system. However, without a standardized way to explain and notate our perceptions, our communication of this information becomes arbitrary and ineffective when designing to display information on mobile interfaces.

Display of Information

- In addition to creating descriptions of our perceptions, we have also standardized a way to classify them. Common classifying schemes that we use are:
- Nominal
  - Uses labels and names to categorize data
- Ordinal
  - Uses numbers to order things in sequence
- Ratio
  - A fixed relationship between one object and another using a zero value as a reference

Display of Information

- Interval
  - The measurable gap between two data value
- Alphabetical
  - Uses the order of the alphabet to organize nominal data
- Geographical
  - Using location, such as city, state, country, to organize data.
  - Uses location, such as city, state, and/or country, to organize data
- Topical
  - Organizing data by topic or subject. Organizes data by topic or subject
- Task
  - Organizing data based on processes, tasks, functions and goals.

Control and Confirmation

- Confirmation,
- Sign On,
- Exit Guard,
- Cancel Protection,
- Timeout

Cognitive Processing

- We are limited in our cognitive processing abilities which are constrained by capacity and duration.

- We have physical limits in our endurance and strength. We have ergonomic limits in our reach and rotation.

- We have perceptive limits in what certain electromagnetic and mechanical wavelengths we can detect and filter.

- Mixed together with our limitations, we expend a lot of cognitive energy to process and interact with the enormous amounts of stimuli in our environment.

Technology Assisted Cognizance

- Let's distribute all cognitive load onto technology.

- What if your refrigerator monitors your shopping habits and cooking behaviors, and can automatically sense which ingredients you need?
  - Then it sends a grocery list order via SMS to your local supermarket.
  - You mobile device can confirm your order was placed, the amount can be charged to your bank account, and you can be notified when your order is ready to be picked up!

Control and Confirmation

- Cognitive frameworks are presented to help us understand how people process information.
- These frameworks apply when interactions require control and confirmation:

Control

- This refers to respecting user data and input while protecting against human error, data loss, and unnecessary decision points. This is a key principle of mobile design.

Control and Confirmation

- Confirmation
- When a necessary decision point is needed, an actionable choice is modally presented to the user to prevent human error. Before adding modal confirmations, consider the following:
  - Is a decision point required here that needs confirmation from the user?
  - Will having this confirmation eliminate risk of human error and loss of input data?

Control and Confirmation Pattern

Confirmation

- When a decision point is reached within a process where the user must confirm an action, or choose between a small number of disparate (and usually exclusive) choices.

Sign On

- This pattern is used to confirm that only authorized individuals have access to a device, site, service, or application on the device.

Control and Confirmation Pattern

Exit Guard

This pattern is used when exiting a screen, process, or application could cause a catastrophic loss of data, or a break in the session.

Cancel Protection

This pattern is used when entered data or subsidiary processes would be time-consuming, difficult, or frustrating to reproduce if lost due to accidental user-selected destruction.

Control and Confirmation Pattern

Timeout

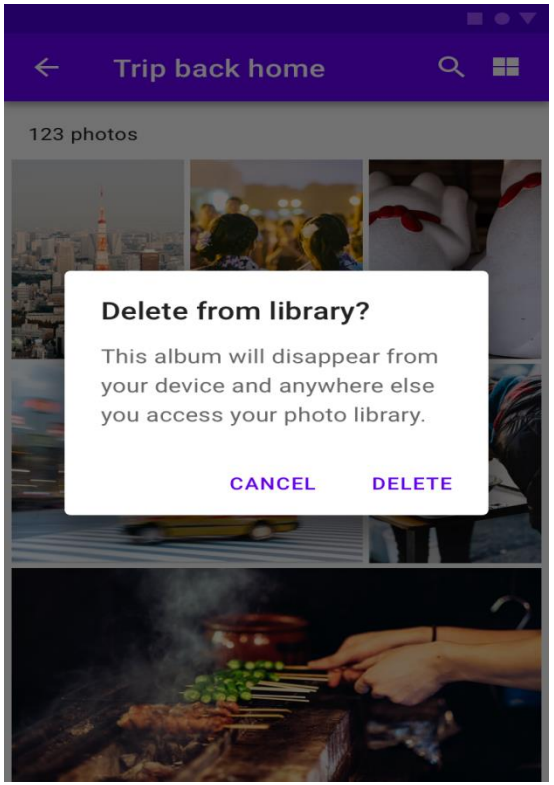
High-security systems or those which are publicly accessed and are likely to be heavily shared (such as kiosks) must have a timer to exit the session and/or lock the sys- tem after a period of inactivity.

Confirmation

- **Confirmation** steps are simple, logical parts of many processes, and can be easily implemented in any number of ways. These patterns discuss the best ways to do so on mobile devices.

- Whenever possible, you should use information from current and previous user behavior, sensors and any other sources, to try to present the correct option to the user. When that is not possible, or likely, present the most likely choice, and give options to switch to others.

Confirmation Types



Single Choice

Confirmation

Three key variations exist: A **single choice** is only used to inform the user.

For example, a fatal error has occurred, and the handset must be restarted. Instead of just doing it for them without warning, it may be helpful to tell the user that the condition has been reached, so you should present a **Confirmation** dialogue, with a single "Restart" button.

A variant of the single choice includes a Wait Indicator to denote that a process -- usually a user-initated one -- has begun and they must wait to perform other tasks. When possible, include a "Cancel" button.

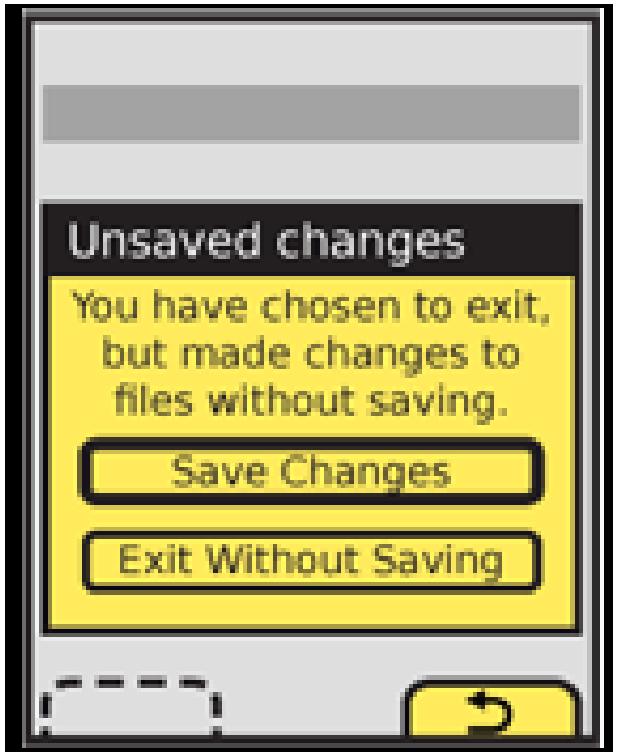
Confirmation Types

Wait Indicator

Confirmation

- By far the most common is the "two or three choices" variant. This is the maximum number of choices that a user can readily comprehended at a glance, while also presenting enough options for most required decisions.
- Examples of three choices are when exiting with unsaved documents:
  - Keep working on the document
  - Save the document and exit
  - Exit without saving the document

Confirmation Messages

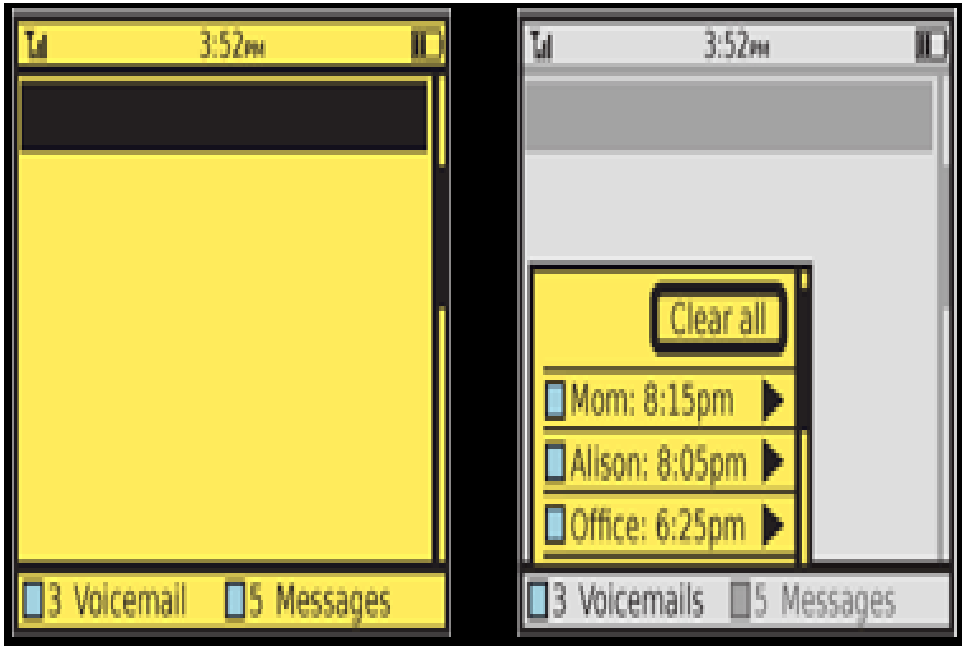


"Save changes"

"Exit without save"

"Cancel"

Confirmation Types



Confirmation – Anti Pattern

- Reduce clicks whenever possible. When a selection is made, it should immediately commit the change and proceed to the next step. Never force the user to make the choice, then press some sort of "Submit" button.

- Do not over use the "single choice" variation. For example, if an application must quit and a **Confirmation** dialogue is useful, a dual choice would usually be better. Offer options to just quit or to quit and restart the same application immediately.

- Carefully consider whether a **Confirmation** dialogue is the right pattern any time more than three choices are offered. It may be a natural part of the process. The use of the modal dialogue implies exceptional conditions or errors so should only be used when needed.

Sign On

Authentication is built into the security model of most devices, but you may want or need to provide additional security for any application, service or site accessed from the device.

You must provide method to allow only authorized individuals access to a device, or a site, service or application on the device

Sign On

- For small, personal-sized devices like phones, MIDs, and tablets, you should generally not obscure or "mask" passwords.
- This is the common practice of replacing the characters with dots or asterisks.
- "Shoulder surfing" on mobiles is extremely difficult due to the scale and viewing angles.
- In addition, the user may be relied on to move to a more private area or simply turn around, behaviors that desktop users cannot exercise.
  - Displaying the entire field will greatly speed entry and reduce errors on entry, all of which not only reduce user frustration, but increase security by reducing time on entry and reducing use of recovery methods.

Sign On – Password Entering

- When the password field may be on the page for a while -- such as account creation -- the password may be obscured when the entire field is not in focus, or after a brief time. You can also give the user a selection, such as a checkbox, to hide the password. You may appreciate this during demonstrations, so you may hide it while projecting your cool new product at a conference.

Sign On – Password Entering

- An excellent method to enter a passcode, for touch and pen devices, is to provide a custom keypad. This may use conventional characters, such as numbers, or special symbols to add a layer of obscurity.
- These are most used as an increased security method; the order of the items changes each time, preventing observers from determining the passcode string by looking at finger patterns or examining the screen for marks.

Sign On – Anti Pattern

- If your use case demands hidden fields, use caution when coding or specifying. Do not assume that the default password method built into the OS is secure.

- Most still reveal the character being typed, and this is basically required in triple-tap text entry modes. If an obscured entry is required due to use as a kiosk, while attached to a projected device or in other shared situations, be sure to specify how obscured it must be. This may require custom fields to be coded.

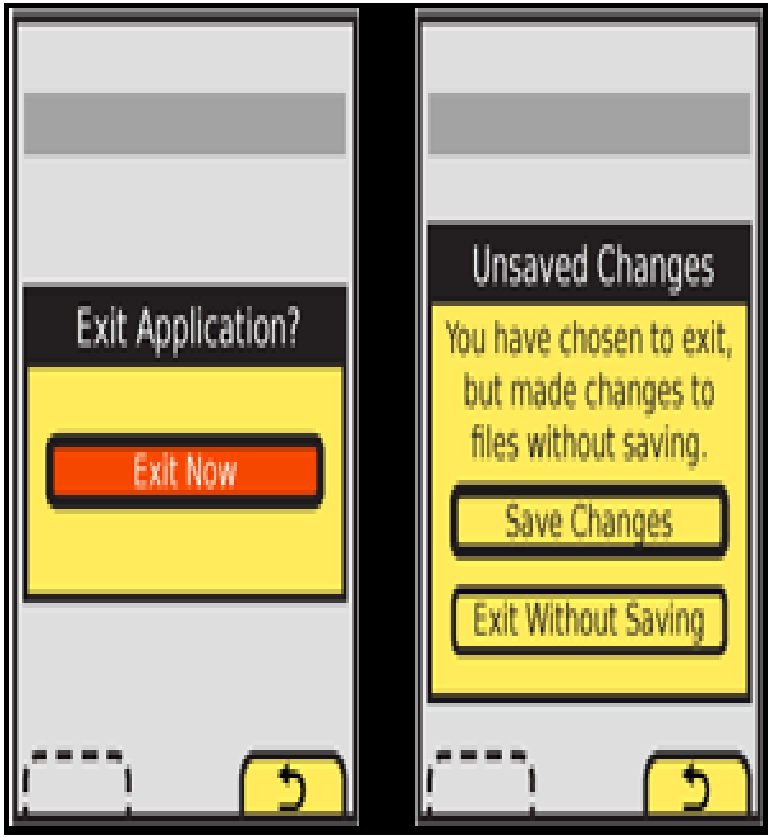
Sign On – Anti Pattern

- Avoid using any desktop paradigms without considering the consequences. For example, do not use dual fields to attempt to solve the entry problems associated with obscured entry.
  - The difficulty of entry on a mobile will make a large percentage of users simply incapable of creating a password with this method.
- Be careful to use terminology absolutely consistently throughout the process. If all your documentation refers to a "passcode" (because it is a number) do not suddenly use the term "Password" on the Sign On screen.

Exit Guards

Exit guards can be built into almost any process, though in some cases preservation of information can be difficult. Plan carefully to assure that the goal of keeping user data is well designed.

For high security or time-sensitive entry, such as banking, you may find it implausible to auto-save information with sufficient security, or a break in the session may cause too much delay. In these cases, an explicit Exit Guard is useful.

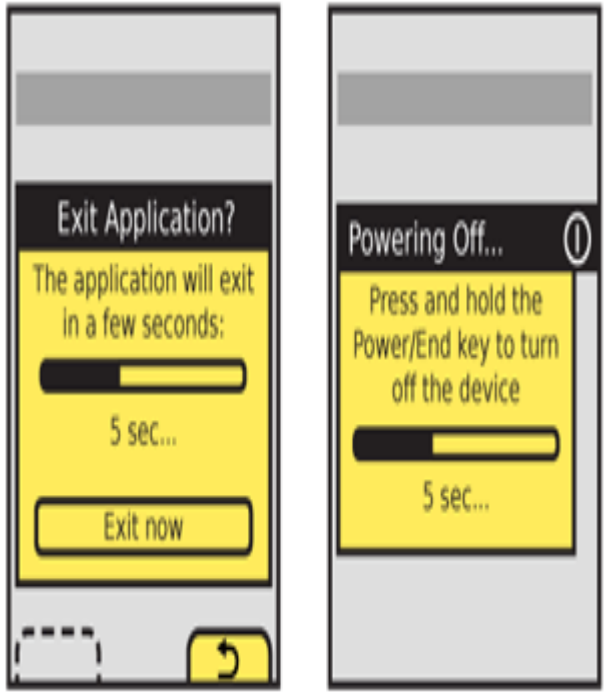


Exit Guards

Timed **Exit Guard** dialogues must notify the user using tones and/or vibration, to assure they are seen.

This may also be useful for conventional **Exit Guard** dialogues, to assure the user does not assume the application or process exited, and then does not check back for some time.

Use the system set volume or vibration settings in general, but always at least vibrate, even if set to no output at all.



Exit Guards – Anti Pattern

- Do not use Exit Guard for all applications and processes. Limit to only those that must be explicitly protected.
- Use only a single dialogue, with a single set of selections.
- Do not make exiting a process, where step one confirms exit, step two is deciding what to do with unsaved documents, and so forth.

Cancel Protection

- All the processes you design must protect or preserve user input.
- You have to provide methods to recover previous and historical entry.
- This is a broad classes of methods, and many of them have no explicit user interface, but instead simply offer the data at the right time.

Cancel Protection

- **Implicit protection:** Design interactive methods to avoid exit or deletion. Take the example of deleting characters in a form field.
- **Repopulation:** When re-entering a previously-used but abandoned form, pre-populate the last-entered information.
- **Explicit protection:** When a single function is provided to clear user entry, provide a method within the screen to allow recovery of the user-entered data.

Cancel Protection

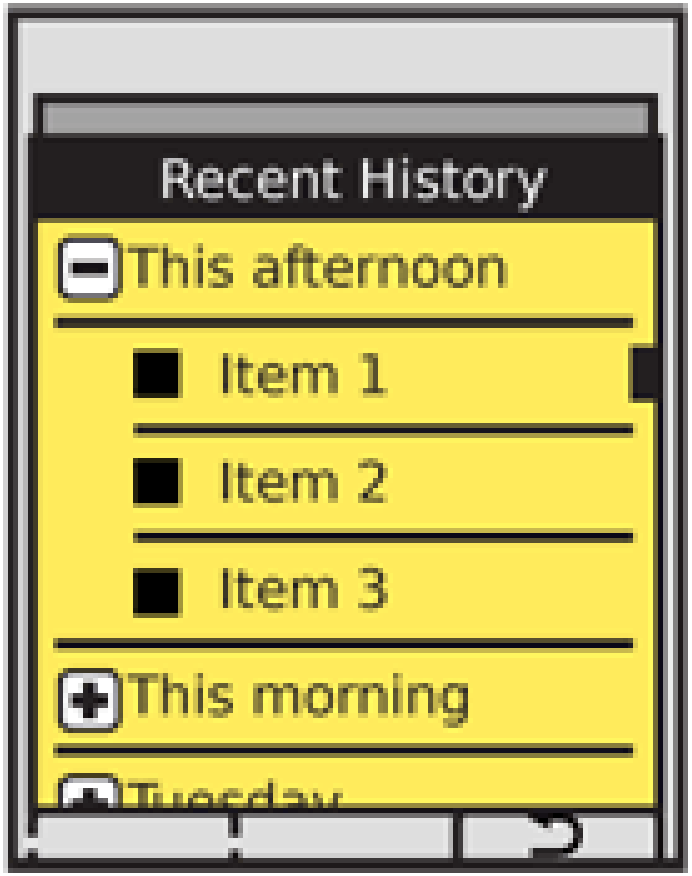
- **Autocomplete & Prediction** processes should save user entries as they are being typed, so they are available for autocomplete even when accidentally deleted or an accidental loss-of-session (such as loss of signal) occurs.
- When entry into fields may be tedious, repetitive or recovered user entry is available, display the autocomplete list of options as soon as a match is found with the current entry.
- To avoid over-using autocomplete, do not attempt to match until a few characters have been typed. The exact number will vary by type of entry, and processing capacity of the device.

Cancel Protection

Cancel Protection

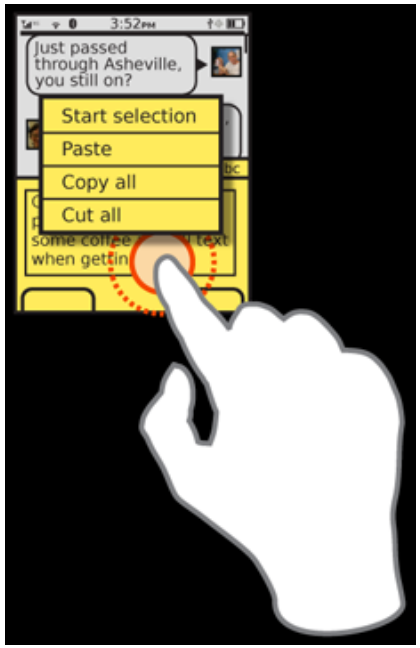
- Implicit protection methods are basically invisible to the user, and if you repopulate a form with previously entered data it usually does not need an explanation.
  - For details on general Autocomplete & Prediction presentation, see that pattern. For the purposes of recovering user information, an indicator should also differentiate user-entered information vs. community or spell-check results, if several types are offered.
  - Displaying history with a Hierarchical List is essentially as described in that pattern. Parents should be labeled by the date and time the session occurred, using natural language to account for ranges; "Yesterday afternoon" is more clear than "1:20-3:45 pm, 7 November."

Cancel Protection



Cancel Protection

Clear Entry functions are built in to some form elements, in some platforms, but in others (such as some web browsers) may be difficult to implement due to restrictions on refreshing data without replacing the entire contents of the page.

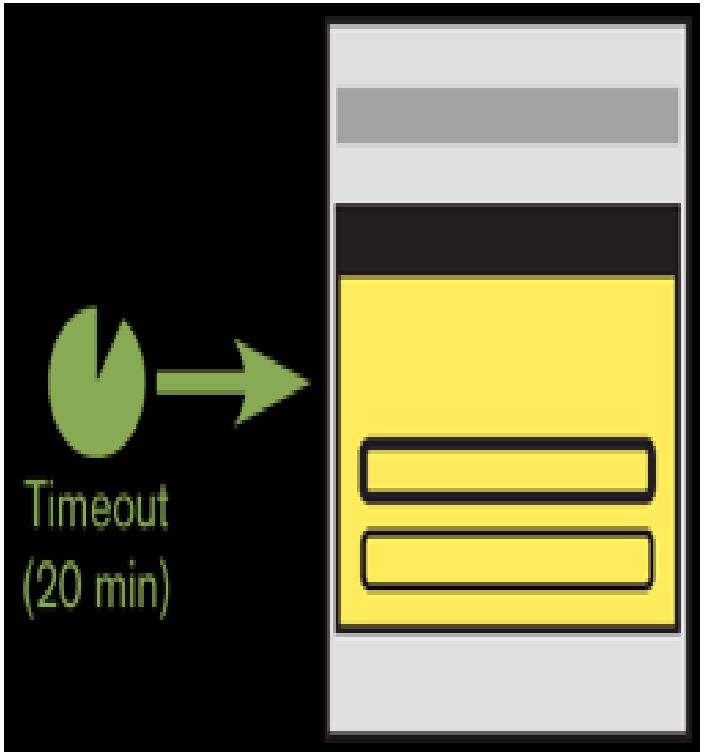


Cancel Protection – Anti Pattern

- Do not preserve secure information such as passwords and financial transaction information without informing the user.

- Do not store any information as plain text that can be searched remotely or when stored as backup files, and do not keep all information forever.

Timeout



When designing high security systems or those which are publicly accessed and are likely to be heavily shared (such as kiosks), you must include a timer to exit the session, or lock the entire system after a period of inactivity.

Timeout

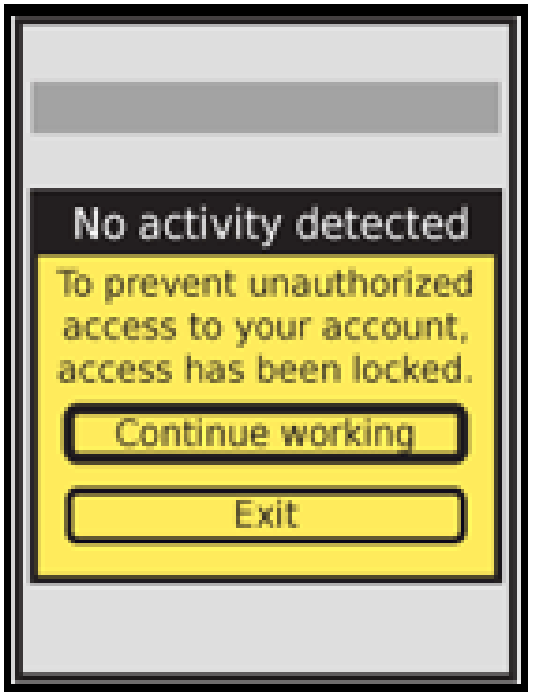
- Many websites or web services use session timeout as a cost-saving measure, to reduce load.
- Affinity based load balancing is the default due to simplicity with sharing user data, but requires session-startup processes or leaving overhead for all predicted users for a particular server.
- Mobile interaction, even with websites, may be interrupted more than desktops so should avoid such behaviors.
- There are other methods, so if system and data design is within scope of your project, consider these as part of the overall experience design.

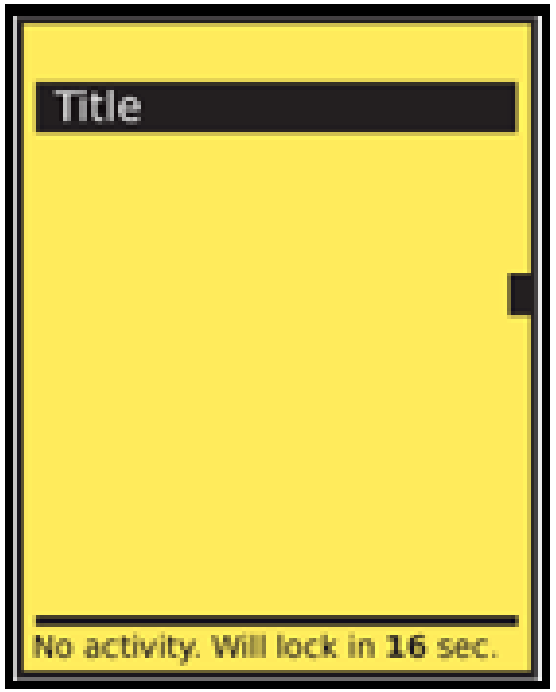
Timeout

- Timeout is a very critical condition, so must be very clear. Anything less intrusive than a modal dialogue often will not be seen.
- The Pop-Up dialogue must be titled and labeled clearly, but with as little technical jargon as possible. Describe the situation not as "timeout" but describe the specific reason such as "Locked due to inactivity," or "Since you walked away…" Emphasize security, and never admit other issues such as load, or discuss a session.
  - Button labels must not depend on the description or title. Never present selections such as "Yes" and "No." At the same time, do not let options be too long, or they may not be readable, or may not look like selections. Typical options are along the lines of:
  - "Continue working"
  - "Exit application"

Samaya TRUST

Timeout


No activity detected
To prevent unauthorized access to your account, access has been locked.
Continue working
Exit


Title
No activity. Will lock in 16 sec.

Timeout


Session Timeout
Your online session will expire in
1 min 2 secs
Please click "Continue" to keep working;
or click "Log Off" to end your session now.
Continue
Log Off

Timeout Anti Pattern

- Do not let system or load calculations determine acceptable timeout. Often, this will be the driving requirement after all, but fight for a user-based method of determining the correct timing.
- Do not use desktop security models, or follow the practices of other products, to determine timeout for mobile or kiosk-based systems.
- Do not ever permanently expire the session without warning the user at the time the session expires