SOFTWARE ENGINEERING

User Interface Design

Interface Design

Typical Design Errors

lack of consistency

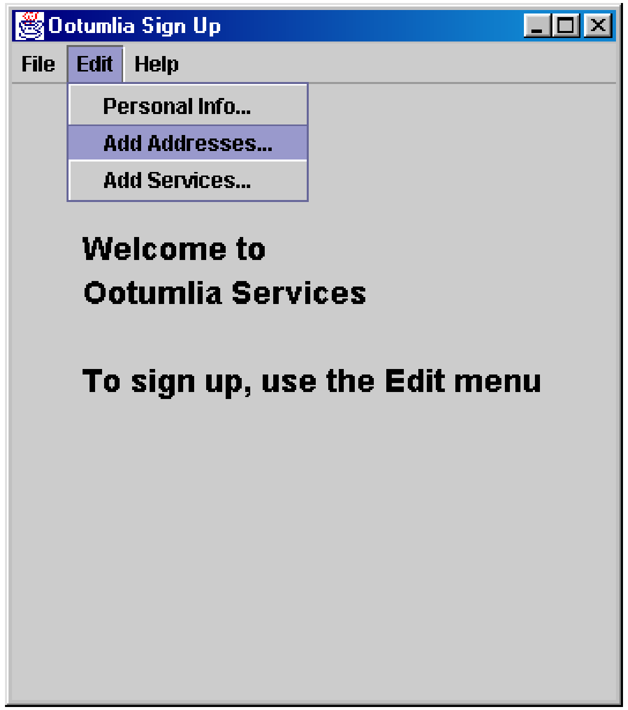
too much memorization

no guidance / help

no context sensitivity

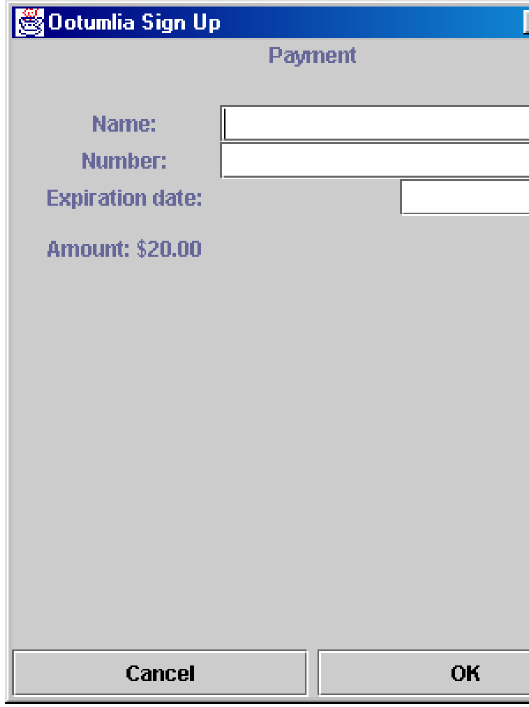
poor response

Arcane/unfriendly



Welcome to
Ootumlia Services

To sign up, use the Edit menu



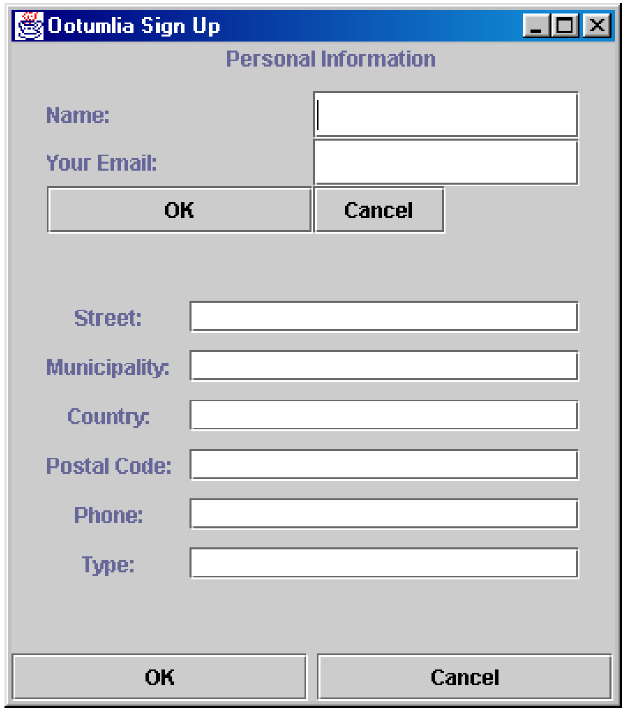
Payment

Name:

Number:

Expiration date:

Amount: $20.00



Personal Information

Name:

Your Email:

Street:

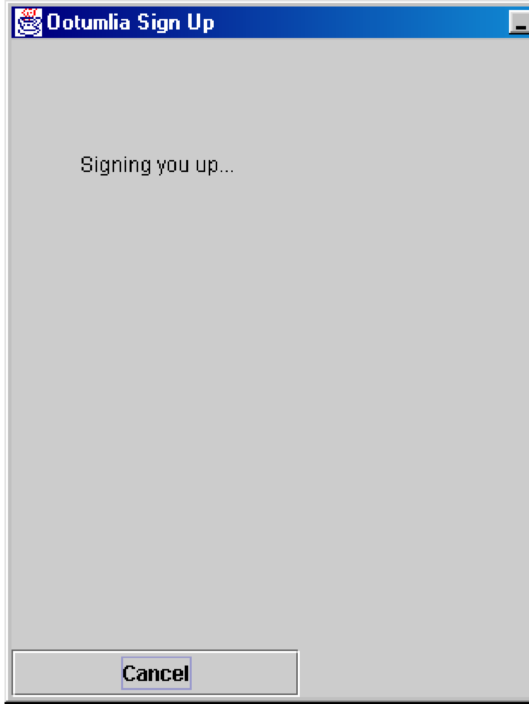
Municipality:

Country:

Postal Code:

Phone:

Type:



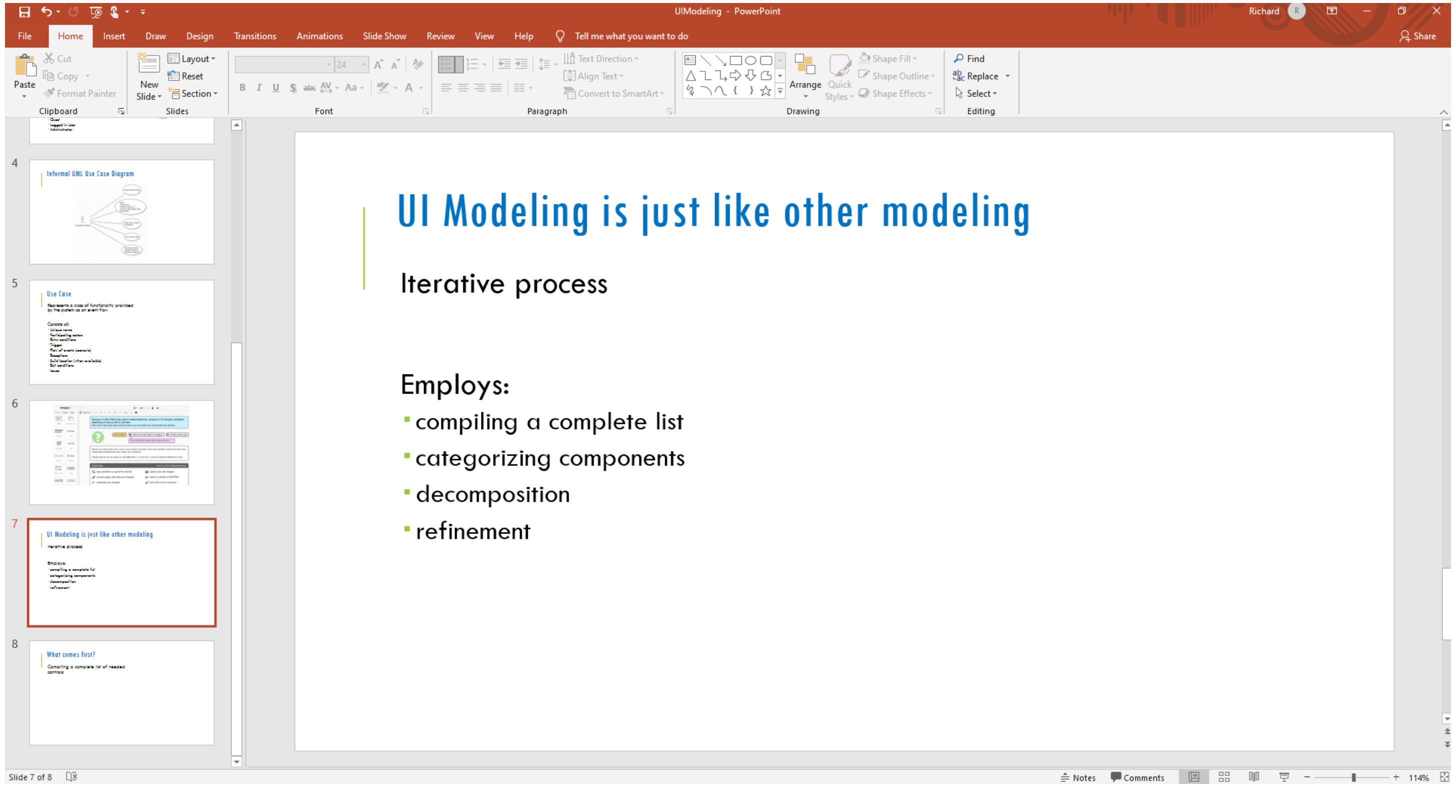
Signing you up...

UI Modeling is just like other modeling

Iterative process

Employs:

- compiling a complete list
- categorizing components
- layout
- design
- refinement



UI Modeling is just like other modeling

Iterative process

Employs:

- compiling a complete list
- categorizing components
- decomposition
- refinement

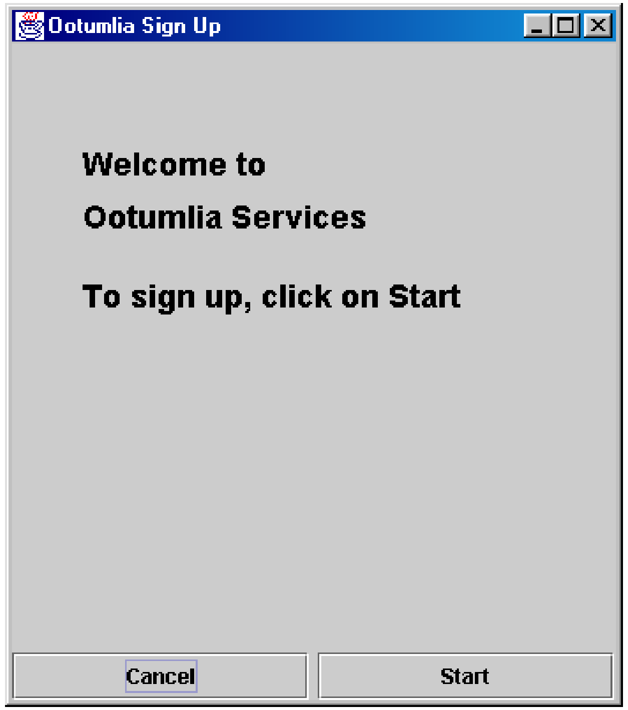
Must be integrated with additional planning

Golden Rules

Place the user in control

Reduce the user's memory load

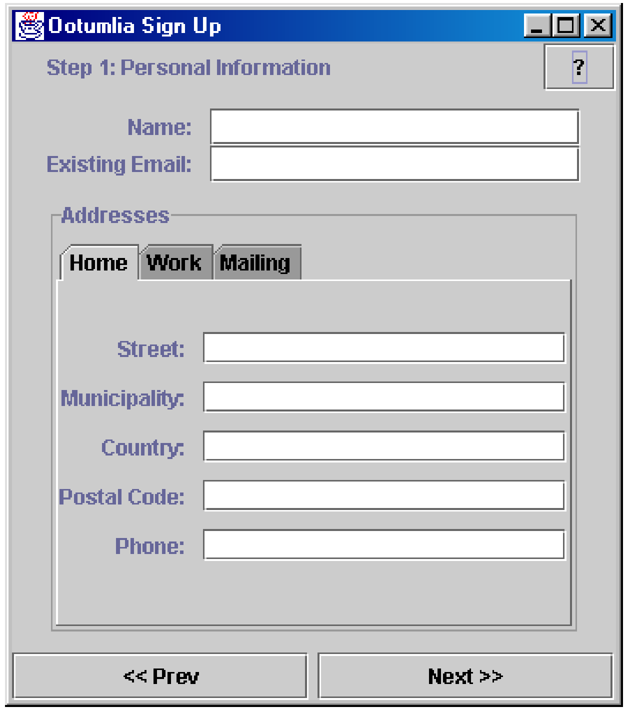
Make the interface consistent


Ootumlia Sign Up

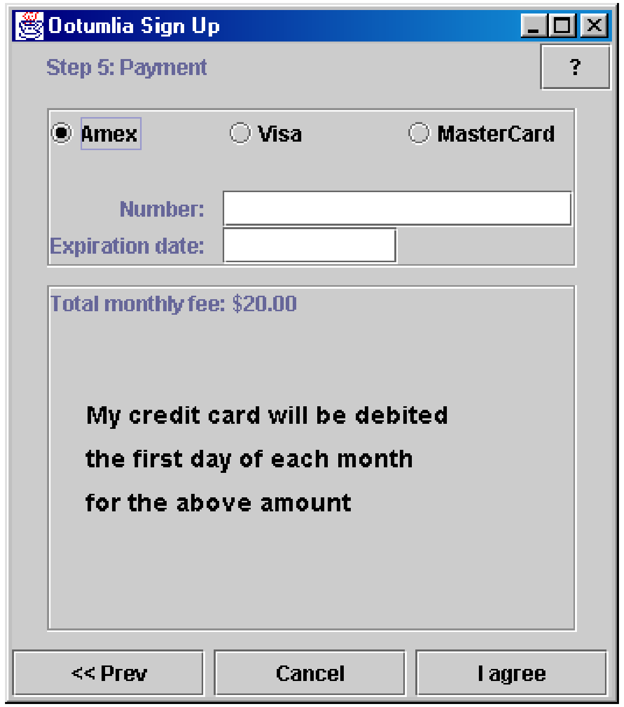
Welcome to

Ootumlia Services

To sign up, click on Start


Ootumlia Sign Up

Step 1: Personal Information

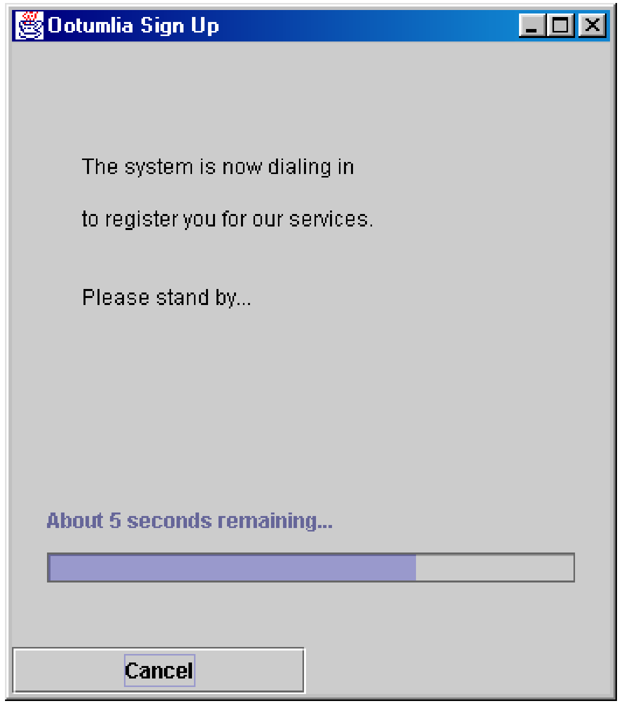

Ootumlia Sign Up

Step 5: Payment

● Amex ○ Visa ○ MasterCard

Total monthly fee: $20.00

My credit card will be debited
the first day of each month
for the above amount


Ootumlia Sign Up

The system is now dialing in

to register you for our services.

Please stand by...

About 5 seconds remaining...

Place the User in Control

- Define interaction modes in a way that does not force a user into unnecessary or undesired actions.
- Provide for flexible interaction.
- Allow user interaction to be interruptible and undoable.
- Streamline interaction as skill levels advance and allow the interaction to be customized.
- Hide technical internals from the casual user.
- Design for direct interaction with objects that appear on the screen.

Reduce the User's Memory Load

- Reduce demand on short-term memory.
- Establish meaningful defaults.
- Define shortcuts that are intuitive.
- The visual layout of the interface should be based on a real world metaphor.
- Disclose information in a progressive fashion.

Make the Interface Consistent

- Allow the user to put the current task into a meaningful context.
- Maintain consistency across a family of applications.
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.

User Interface Analysis &Design Models

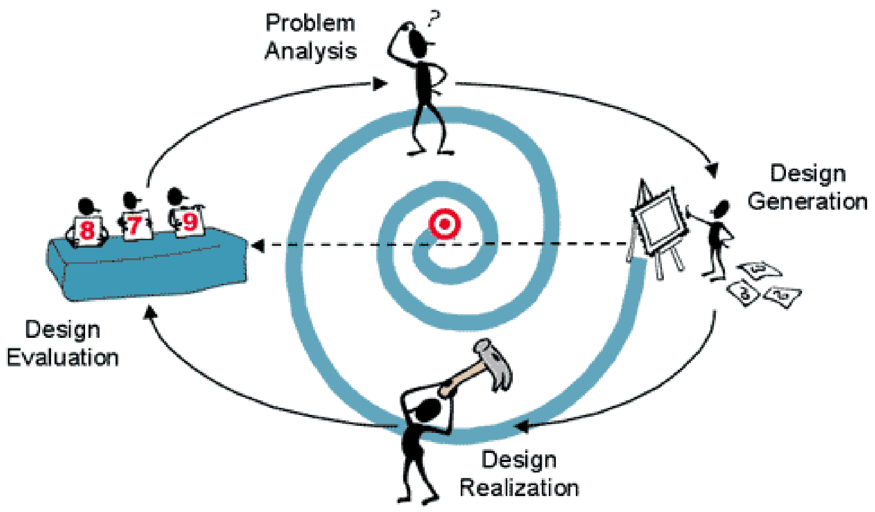
User model — a profile of all end users of the system

Design model — a design realization of the user model

Mental model (system perception) — the user's mental image of what the interface is

Implementation model — the interface "look and feel" coupled with supporting information that describe interface syntax and semantics

A Spiral Process – UI Development



Interface analysis (user, task, and environment analysis)

- Focuses on the profile of the users who will interact with the system
- Concentrates on users, tasks, content and work environment

Interface design

- Defines a set of interface objects and actions (and their screen representations) that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system

Interface construction

- Begins with a prototype that enables usage scenarios to be evaluated
- Continues with development tools to complete the construction

Interface validation

- implement every user task correctly, to accommodate all task variations, and to achieve all general user requirements

User Interface Analysis

Elements of the User Interface

- To perform user interface analysis, the practitioner needs to study and understand four elements
  - The users who will interact with the system through the interface
  - The tasks that end users must perform to do their work
  - The content that is presented as part of the interface
  - The work environment in which these tasks will be conducted

User Interface Analysis-User Analysis

The analyst strives to get the end user's mental model and the design model to converge by understanding

- The **users** themselves
- **How** these people **use** the system

Information can be obtained from

- User interviews with the end users
- Sales input from the sales people who interact with customers and users on a regular basis

A set of questions should be answered during user analysis

- Are the users trained professionals, technicians, clerical or manufacturing workers?
- What level of formal education does the average user have?
- Are the users capable of learning on their own from written materials or have they expressed a desire for classroom training?
- Are the users expert typists or are they keyboard phobic?
- What is the age range of the user community?
- Will the users be represented predominately by one gender?

User Interface Analysis-Task Analysis and Modeling

Task analysis strives to **know** and **understand**

- The work the user performs in specific circumstances
- The tasks and subtasks that will be performed as the user does the work
- The specific problem domain objects that the user manipulates as work is performed
- The sequence of work tasks (i.e., the workflow)
- The hierarchy of tasks

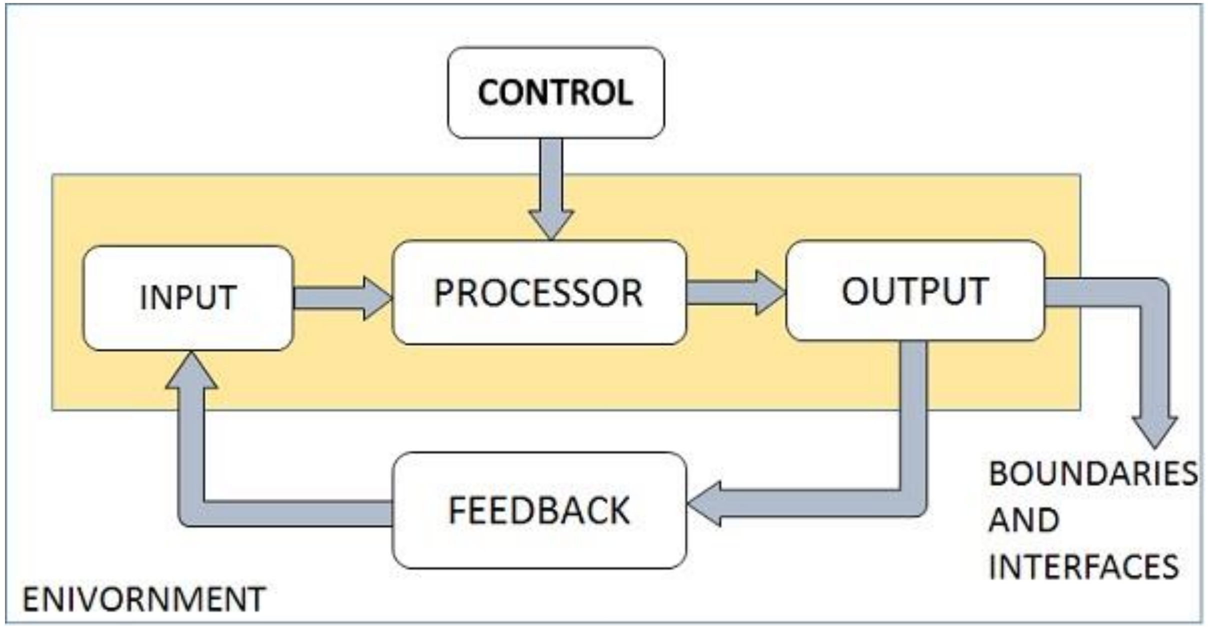
Use cases

- Show how an end user performs some specific work-related task
- Enable the software engineer to extract tasks, objects, and overall workflow of the interaction
- Helps the software engineer to identify additional helpful features

First compile a complete list

For Use Cases:

- What data must the UI provide?
  - ignore data storage for now
- What data must the UI update?
  - ignore data storage for now
- Compile complete list of controls



User Interface Analysis-Content Analysis

The display content may range from character-based reports, to graphical displays, to multimedia information

Display content may be

- Generated by components in other parts of the application
- Acquired from data stored in a database that is accessible from the application
- Transmitted from systems external to the application in question

The **format** and **aesthetics** of the content (as it is displayed by the interface) needs to be considered

A set of questions should be answered during content analysis:

- Are various types of data assigned to consistent locations on the screen (e.g., photos always in upper right corner)?
- Are users able to customize the screen location for content?
- How is color used to enhance understanding?
- How are error messages and warnings presented in order to make them quick and easy to see and understand?

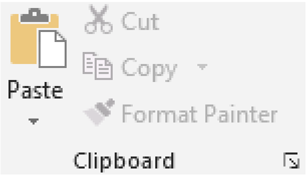
Categorize Components

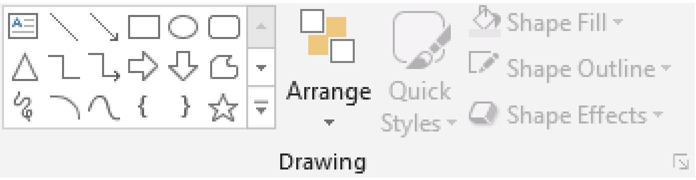
Components, i.e. controls and containers

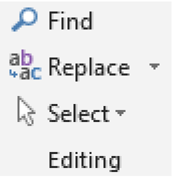
- What actions are they associated with?
- What input do they receive?
- What output do they provide?

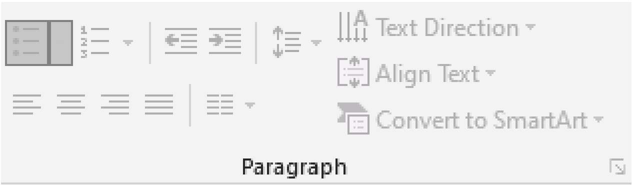
Group them accordingly

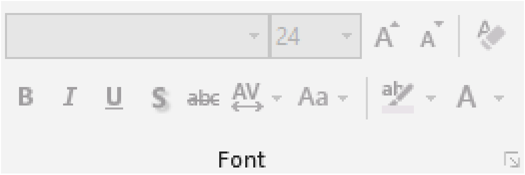
- multistep tasks - forms
- similar tasks - toolbars
- conserve space
- ensure clarity of use

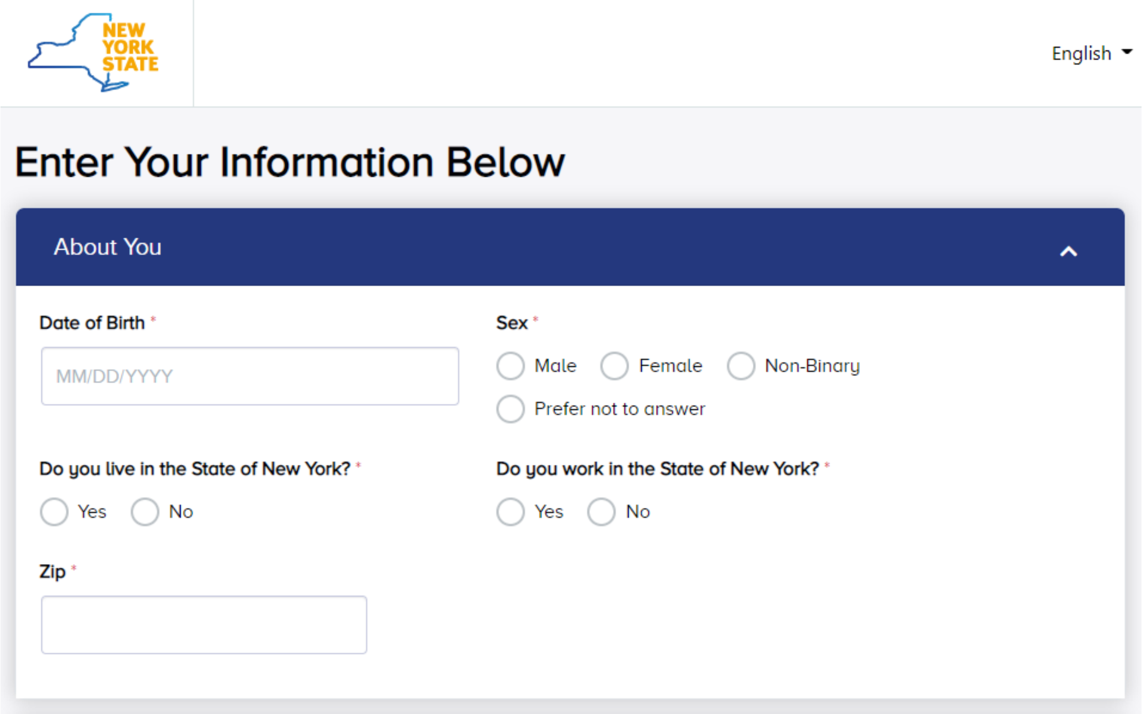












Enter Your Information Below

About You

Date of Birth *

MM/DD/YYYY

Sex *

○ Male ○ Female ○ Non-Binary

○ Prefer not to answer

Do you live in the State of New York? *

○ Yes ○ No

Do you work in the State of New York? *

○ Yes ○ No

Zip *

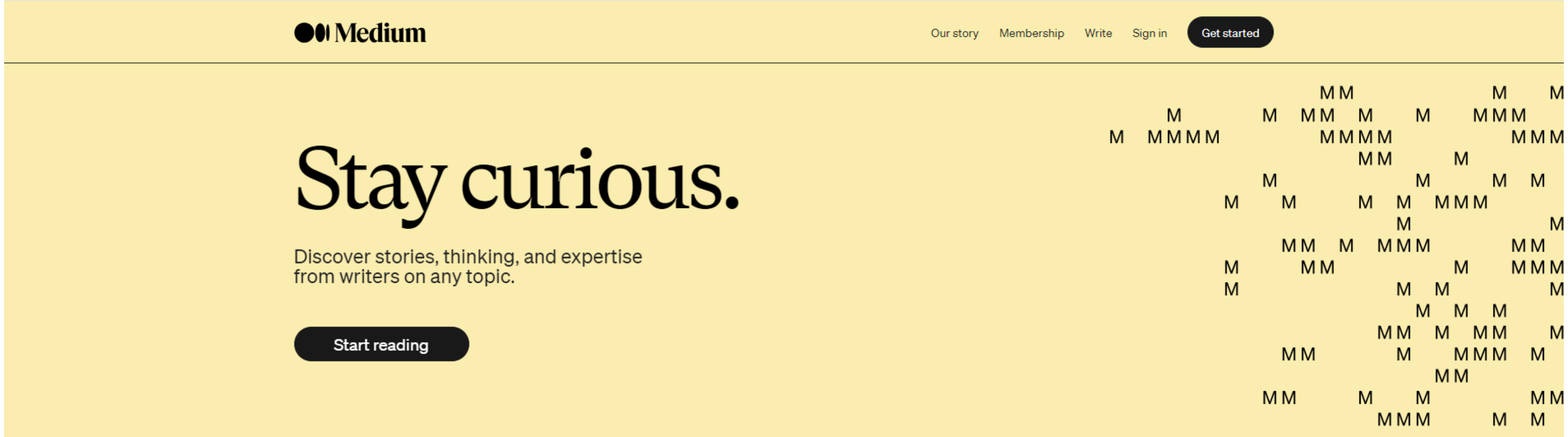
User Interface Analysis-Work Environment Analysis

Software products need to be designed to fit into the work environment, otherwise they may be difficult or frustrating to use

Factors to consider include

- Type of lighting
- Display size and height
- Keyboard size, height and ease of use
- Mouse type and ease of use
- Surrounding noise
- Space limitations for computer and/or user
- Weather or other atmospheric conditions
- Temperature or pressure restrictions
- Time restrictions (when, how fast, and for how long)

Design



Stay curious.

Discover stories, thinking, and expertise from writers on any topic.

Make things look beautiful

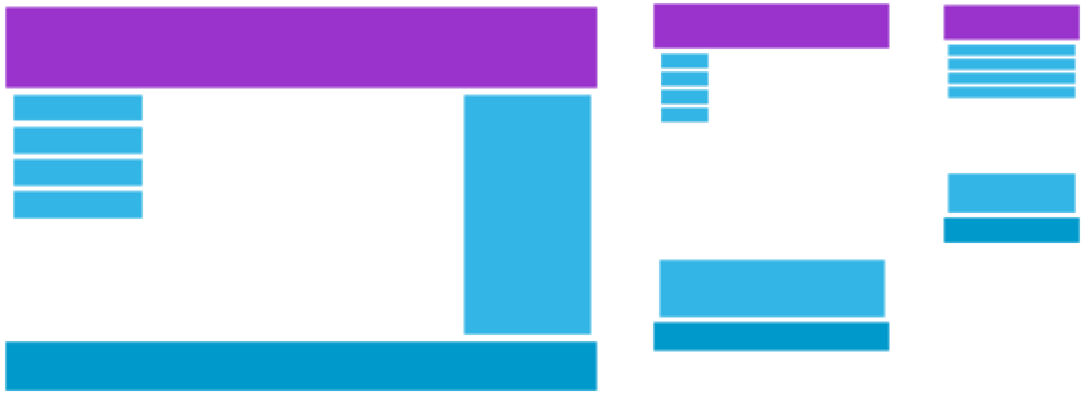
User Interface Design

User interface design is an iterative process, where each iteration elaborate and refines the information developed in the preceding step

General steps for user interface design

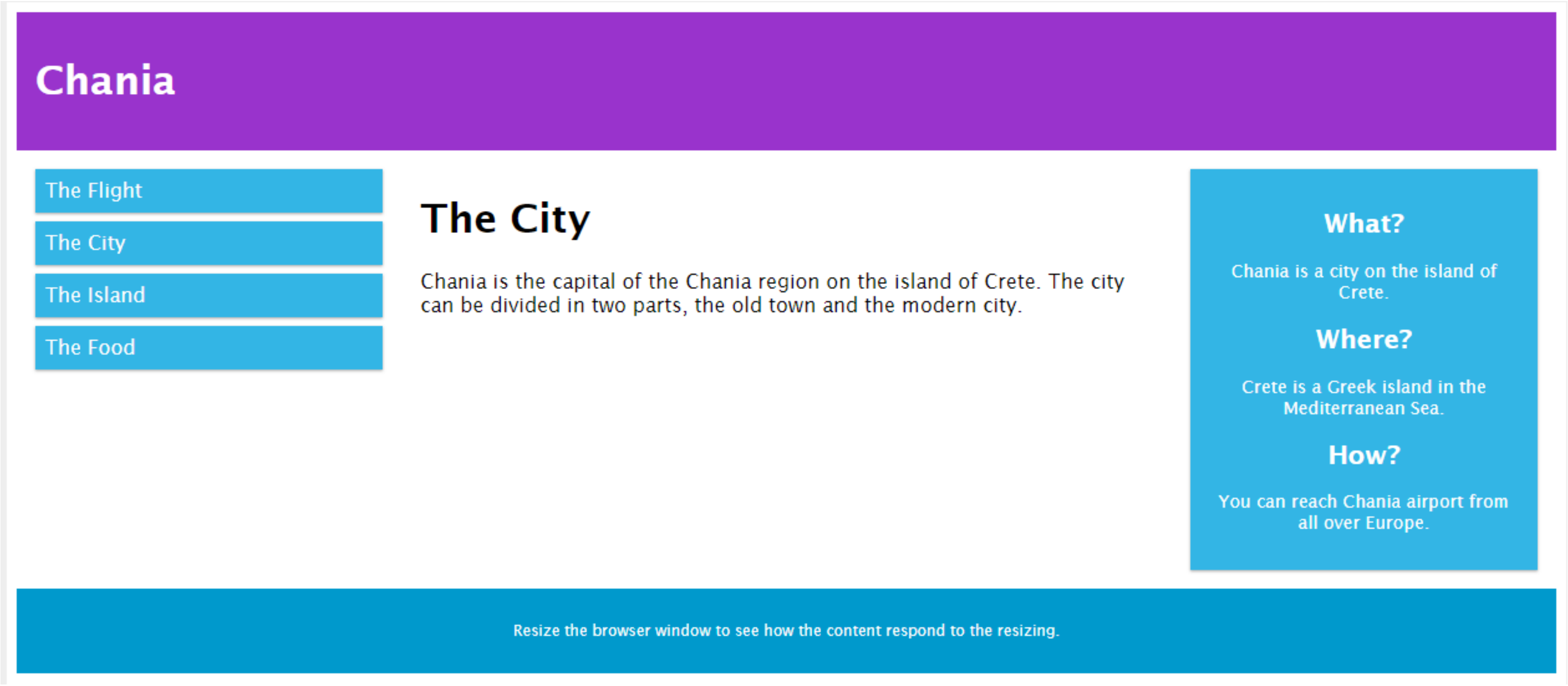
- Using information developed during user interface analysis, **define** user interface objects and actions (operations)
- Define **events** (user actions) that will cause the state of the user interface to change; **model** this behavior
- Depict each interface state as it will actually look to the end user
- Indicate how the user **interprets** the state of the system from information provided through the interface

Layout



Desktop vs Tablet vs Mobile

A way of grouping groups of components



Chania

- The Flight
- The City
- The Island
- The Food

The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

What?

Chania is a city on the island of Crete.

Where?

Crete is a Greek island in the Mediterranean Sea.

How?

You can reach Chania airport from all over Europe.

Resize the browser window to see how the content respond to the resizing.

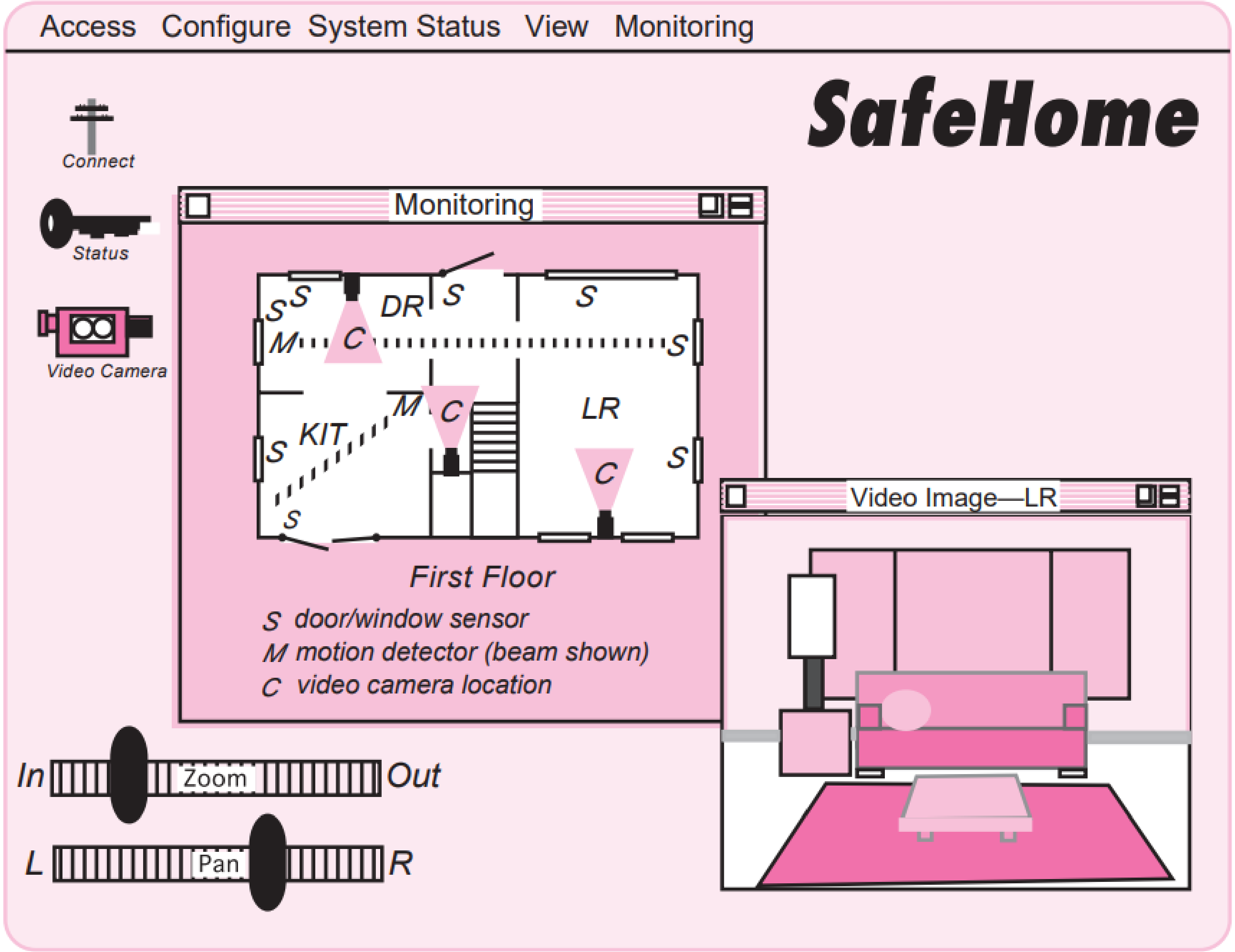
Example Use Case:SafeHome

I want to gain access to my SafeHome system from any remote location via the Internet. Using browser software operating on my notebook computer (while I'm at work or traveling), I can determine the status of the alarm system, arm or disarm the system, reconfigure security zones, and view different rooms within the house via preinstalled video cameras. To access SafeHome from a remote location, I provide an identifier and a password. These define levels of access (e.g., all users may not be able to reconfigure the system) and provide security. Once validated, I can check the status of the system and change the status by arming or disarming SafeHome. I can reconfigure the system by displaying a floor plan of the house, viewing each of the security sensors, displaying each currently configured zone, and modifying zones as required. I can view the interior of the house via strategically placed video cameras. I can pan and zoom each camera to provide different views of the interior

- accesses the SafeHome system
- enters an ID and password to allow remote access
- checks system status
- arms or disarms SafeHome system
- displays floor plan and sensor locations
- displays zones on floor plan
- changes zones on floor plan
- displays video camera locations on floor plan
- selects video camera for viewing
- views video images (four frames per second)
- pans or zooms the video camera

- *accesses* the SafeHome system
- *enters* an **ID** and **password** to allow remote access
- *checks* **system status**
- *arms* or *disarms* SafeHome system
- *displays* **floor plan** and **sensor locations**
- displays **zones** on floor plan
- *changes* **zones** on floor plan
- *displays* **video camera locations** on floor plan
- *selects* **video camera** for viewing
- *views* **video images** (four frames per second)
- *pans* or *zooms* the **video camera**

Screen layout



User Interface Design-User-system Interaction

Two problems must be addressed in **interactive** systems design

- How should information from the user be **provided** to the computer system?
- How should information from the computer system be **presented** to the user?

Design Issues

The design of a user interface evolves, four common design issues almost always surface:

- System response time
- User help facilities
- Error information handling
- Command labeling.
- Application accessibility
- Internationalization.

User Interface Design-Guidelines for Error Messages

The message should

- describe the problem in plain language that a typical user can **understand**
- provide **constructive advice** for recovering from the error
- indicate any negative consequences of the error (e.g., potentially corrupted data files) so that the user can check to ensure that they have not occurred (or correct them if they have)
- be accompanied by an audible or visual cue such as a beep, momentary flashing, or a special error color
- be non-judgmental (never place blame on the user)

An effective error message philosophy can do much to improve the quality of an interactive system and will significantly reduce user frustration when problems do occur

User Interface Design-Menu Labeling and Typed Commands

Questions for Menu Labeling and Typed Commands

- Will every menu option have a corresponding command?
- What form will a command take? A control sequence? A function key? A typed word?
- How difficult will it be to learn and remember the commands?
- What can be done if a command is forgotten?
- Can commands be customized or abbreviated by the user?
- Are menu labels self-explanatory within the context of the interface?
- Are submenus consistent with the function implied by a master menu item?

User Interface Design-UI design Principles

UI design must take account of the **needs, experience** and **capabilities** of the system users

Designers should be aware of people’s **physical** and **mental** limitations (e.g. limited short-term memory) and should recognize that people make mistakes

User Interface Design-UI Design Principles

Anticipation

- An App should be designed so that it expect the user's next move.
  - Example: Software installation procedures

Communication

- The interface should communicate the status of any activity initiated by the user.
  - Example: file copying shown via progress bar.

Consistency

- The system should display an appropriate level of consistency.
- Commands and menus should have the same format, command punctuation should be similar, etc.

Controlled autonomy.

- The interface should facilitate user movement throughout the App, but it should do so in a manner that enforces navigation conventions that have been established for the application
  - Enforce id and password for no go options.

User Interface Design-UI Design Principles

Efficiency

- The design of the App and its interface should optimize the user's work efficiency, not the efficiency of the developer who designs and builds it or the client-server environment that executes it.
    - Input the CNIC or home number without space.

Flexibility

- The interface should be flexible enough to enable some users to accomplish tasks directly and others to explore the App in a somewhat random fashion.

Focus

- The App interface (and the content it presents) should stay focused on the user task(s) at hand.

Fitt's law

- Law:"The time to acquire a target is a function of the distance to and size of the target".
    - For Example, Each user option implies a set of follow-on user choices or actions. For example, the "buy a product" option requires that the user enter a product category followed by the product name. The product category (e.g., audio equipment, televisions, DVD players) appears as a pull-down menu as soon as "buy a product" is picked. Therefore, the next choice is immediately obvious (it is nearby) and the time to acquire it is negligible.

User Interface Design-UI Design Principles

Human interface objects

- A vast library of reusable human interface objects has been developed for Apps(such as template).

Latency reduction

- Rather than making the user wait for some internal operation to complete (e.g., downloading a complex graphical image), the App should use multitasking in a way that lets the user proceed with work as if the operation has been completed.

Learnability

- An App interface should be designed to minimize learning time, and once learned, to minimize relearning required when the App is revisited.

Metaphors

- An interface that uses an interaction metaphor is easier to learn and easier to use, as long as the metaphor is appropriate for the application and the user

User Interface Design-UI Design Principles

Maintain work product integrity.

- A work product (e.g., a form completed by the user, a user-specified list) must be automatically saved so that it will not be lost if an error occurs.

Readability.

- All information presented through the interface should be readable by young and old.

Track state

- When appropriate, the state of the user interaction should be tracked and stored so that a user can logoff and return later to pick up where she left off.

Visible navigation.

- A well-designed App interface provides "the illusion that users are in the same place, with the work brought to them"

User Interface Design-UI Design Guidelines

- Reading speed on a computer monitor is approximately 25 percent slower than reading speed for hardcopy. Therefore, do not force the user to read voluminous amounts of text, particularly when the text explains the operation of the WebApp or assists in navigation.
- Avoid “under construction” signs—an unnecessary link is sure to disappoint.
- Users prefer not to scroll. Important information should be placed within the dimensions of a typical browser window.
- Navigation menus and head bars should be designed consistently and should be available on all pages that are available to the user. The design should not rely on browser functions to assist in navigation.
- Aesthetics should never supersede functionality. For example, a simple button might be a better navigation option than an aesthetically pleasing, but vague image or icon whose intent is unclear.
- Navigation options should be obvious, even to the casual user. The user should not have to search the screen to determine how to link to other content or services.

Design and Prototype Evaluation

Before prototyping occurs, a number of evaluation criteria can be applied during design reviews to the design model itself

- The amount of learning required by the users
  - Derived from the length and complexity of the written specification and its interfaces
- The interaction time and overall efficiency
  - Derived from the number of user tasks specified and the average number of actions per task
- The memory load on users
  - Derived from the number of actions, tasks, and system states
- The complexity of the interface and the degree to which it will be accepted by the user
  - Derived from the interface style, help facilities, and error handling procedures

Design and Prototype Evaluation

Prototype evaluation can range from an informal test drive to a formally designed study using statistical methods and questionnaires

The prototype evaluation cycle consists of prototype creation followed by user evaluation and back to prototype modification until all user issues are resolved

The prototype is evaluated for

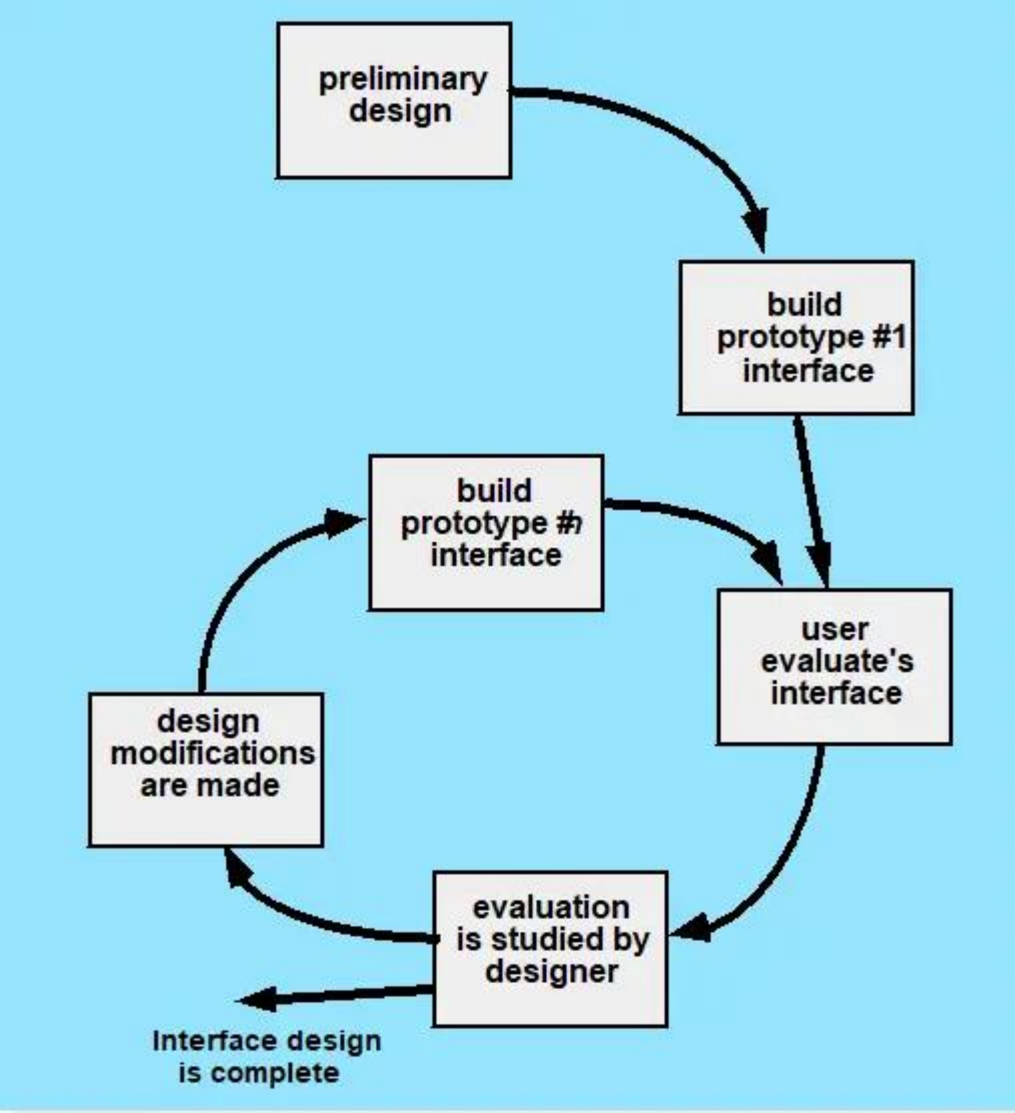
- Satisfaction of user requirements
- Conformance to the golden rules of user interface

Design and Prototype Evaluation-Evaluation Techniques

Simple evaluation techniques

- Questionnaires for user feedback
- Video recording of system use and subsequent tape evaluation.
- Instrumentation of code to collect information about facility use and user errors.
- The provision of a grip button for on-line user feedback.

Design Evaluation Cycle


preliminary design
build prototype #1 interface
build prototype #n interface
user evaluate's interface
design modifications are made
evaluation is studied by designer
Interface design is complete

We want to make beautiful, efficient GUIs

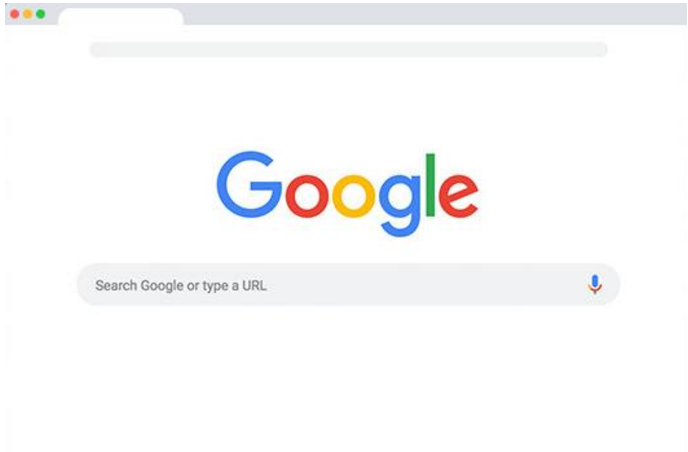
Where do we start?

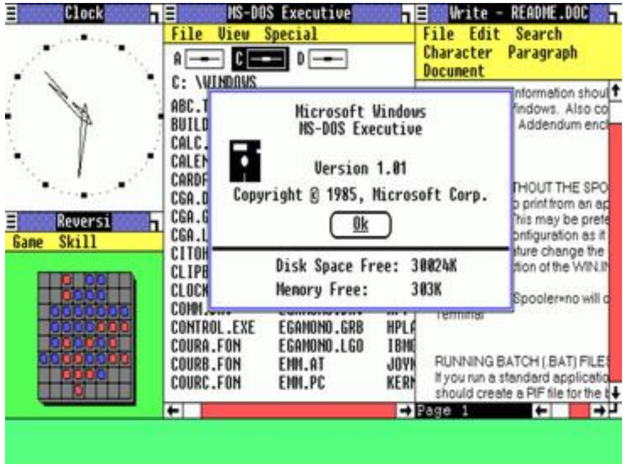
Understand UI design principles

Study available design philosophies

Learn from existing examples

- good and bad



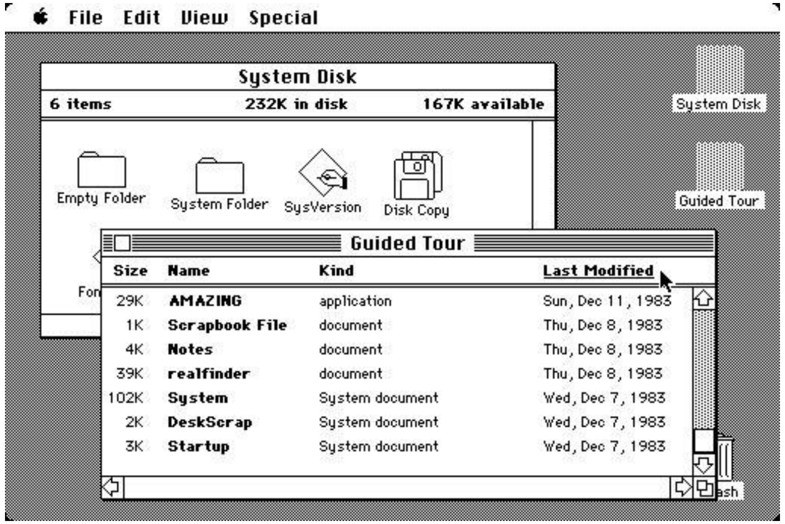


Microsoft Windows
MS-DOS Executive

Version 1.01

Copyright © 1985, Microsoft Corp.

Disk Space Free: 30024K
Memory Free: 303K



6 items 232K in disk 167K available

Size	Name	Kind	Last Modified
29K	AMAZING	application	Sun, Dec 11, 1983
1K	Scrapbook File	document	Thu, Dec 8, 1983
4K	Notes	document	Thu, Dec 8, 1983
39K	realfinder	document	Thu, Dec 8, 1983
102K	System	System document	Wed, Dec 7, 1983
2K	DeskScrap	System document	Wed, Dec 7, 1983
3K	Startup	System document	Wed, Dec 7, 1983

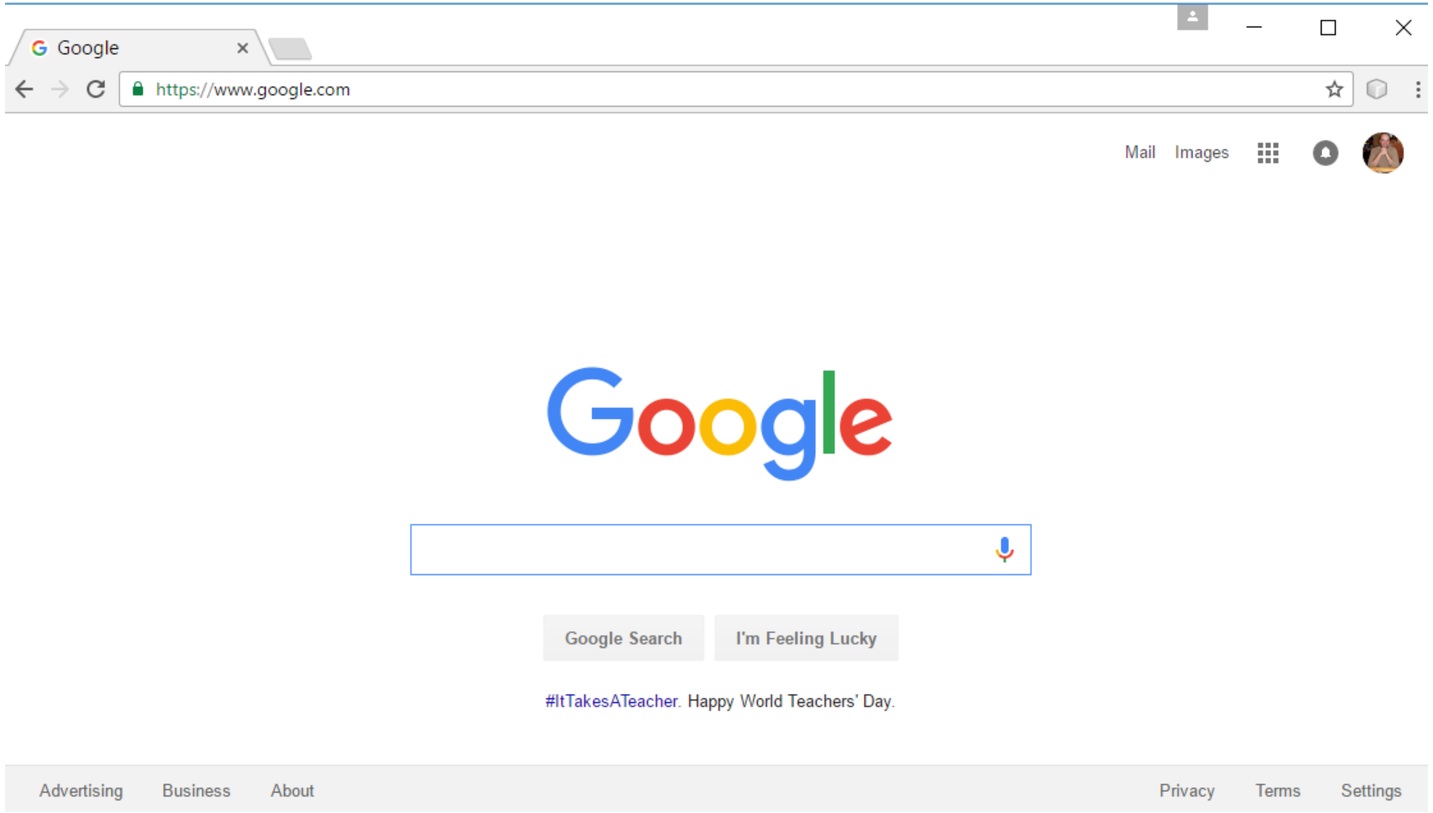
Characteristics of Human Perception

Proximity

Similarity

Balance

Vision



#ItTakesATeacher. Happy World Teachers' Day.

Visually Pleasing Composition

Balance

Symmetry

Regularity

Predictability

Sequentiality

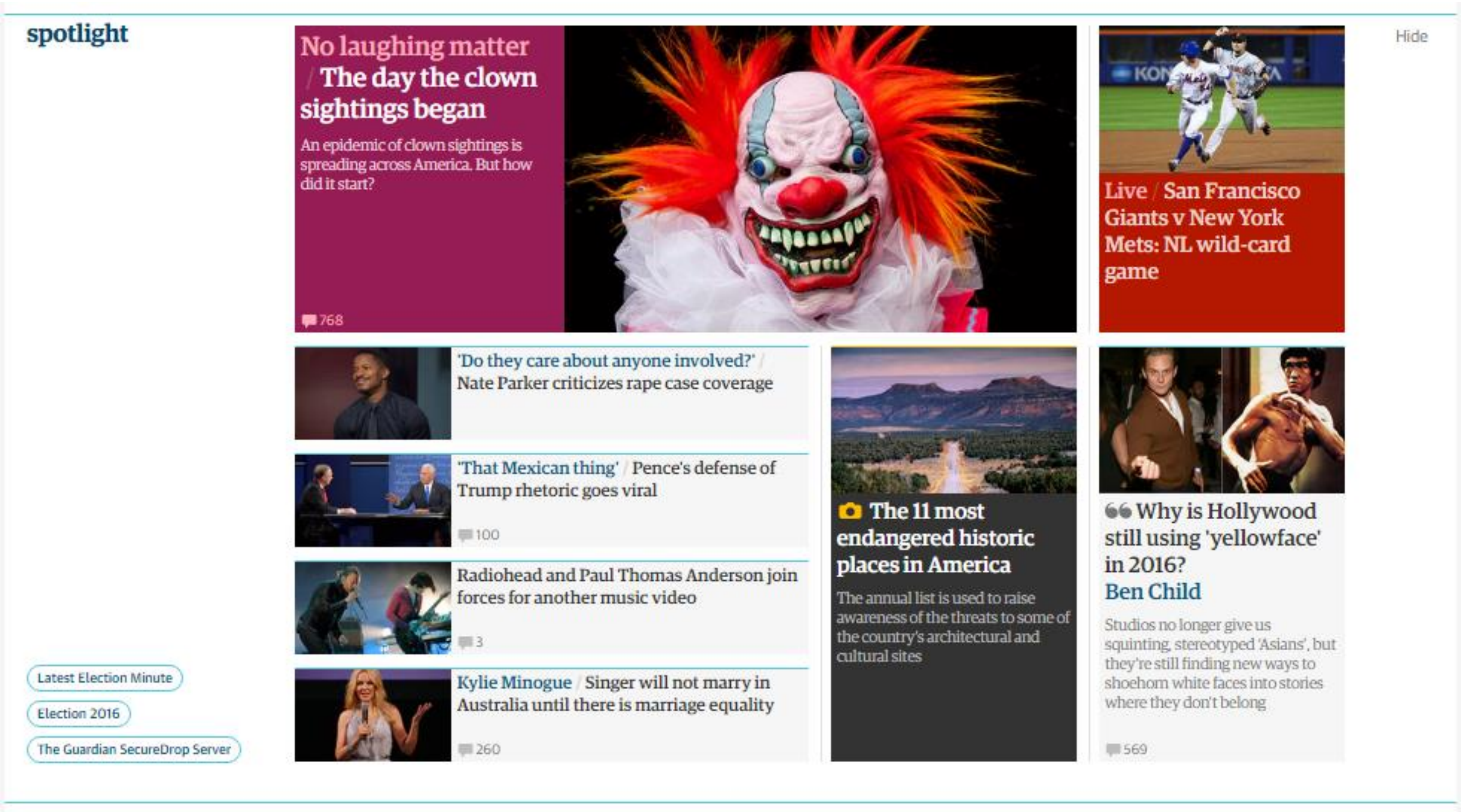
Economy

Unity

Proportion

Simplicity

Groupings



spotlight

Hide

No laughing matter / The day the clown sightings began

An epidemic of clown sightings is spreading across America. But how did it start?

768

Live / San Francisco Giants v New York Mets: NL wild-card game

'Do they care about anyone involved?' / Nate Parker criticizes rape case coverage

'That Mexican thing' / Pence's defense of Trump rhetoric goes viral

100

Radiohead and Paul Thomas Anderson join forces for another music video

3

Kylie Minogue / Singer will not marry in Australia until there is marriage equality

260

The 11 most endangered historic places in America

The annual list is used to raise awareness of the threats to some of the country's architectural and cultural sites

Why is Hollywood still using 'yellowface' in 2016? Ben Child

Studios no longer give us squinting, stereotyped 'Asians', but they're still finding new ways to shoehorn white faces into stories where they don't belong

569

Latest Election Minute

Election 2016

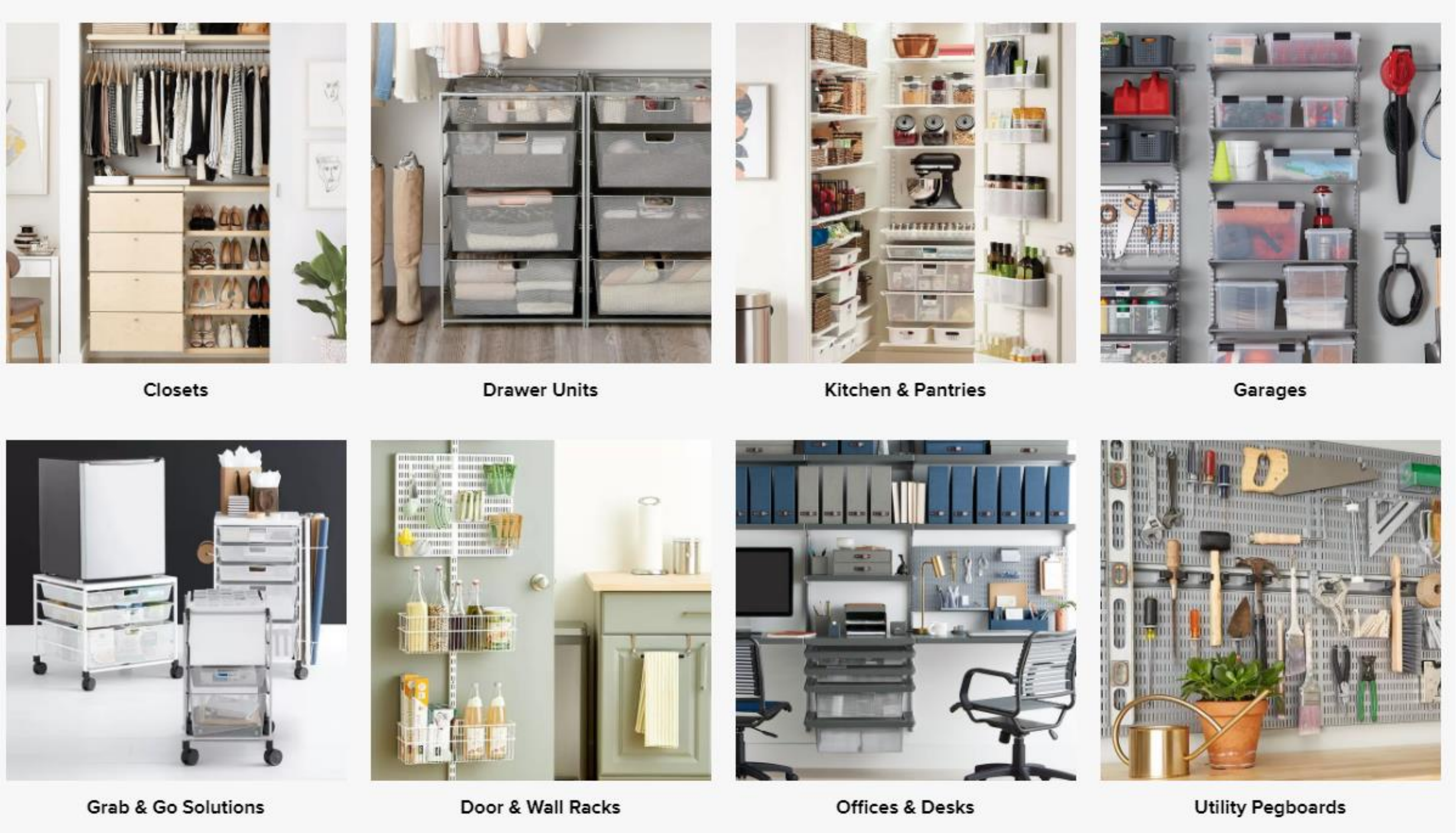
The Guardian SecureDrop Server

One tool to help with all of these things: ***Cards***

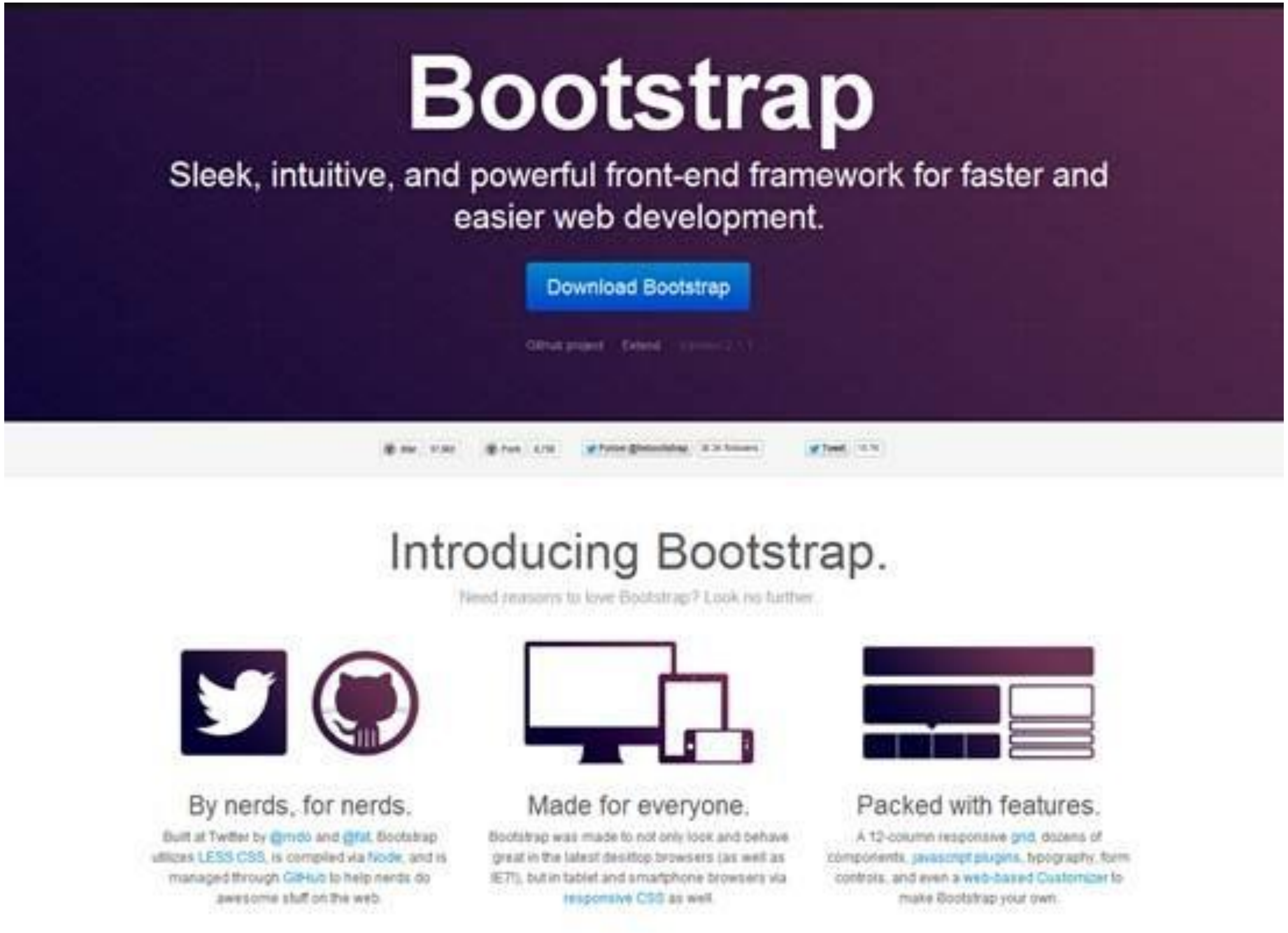
UI Design has its own principles

Alignment, Balance, Clarity, Compatibility, Comprehensibility, Configurability, Consistency, Control, Directness, Efficiency, Familiarity, Flexibility, Forgiveness, Predictability, Recovery, Responsiveness, Simplicity, Uniformity

I love The Container Store. Why?



For example, Bootsstrap



Bootstrap

Sleek, intuitive, and powerful front-end framework for faster and easier web development.

Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.

By nerds, for nerds.

Built at Twitter by @mdo and @fat, Bootstrap utilizes LESS CSS, is compiled via Node, and is managed through GitHub to help nerds do awesome stuff on the web.

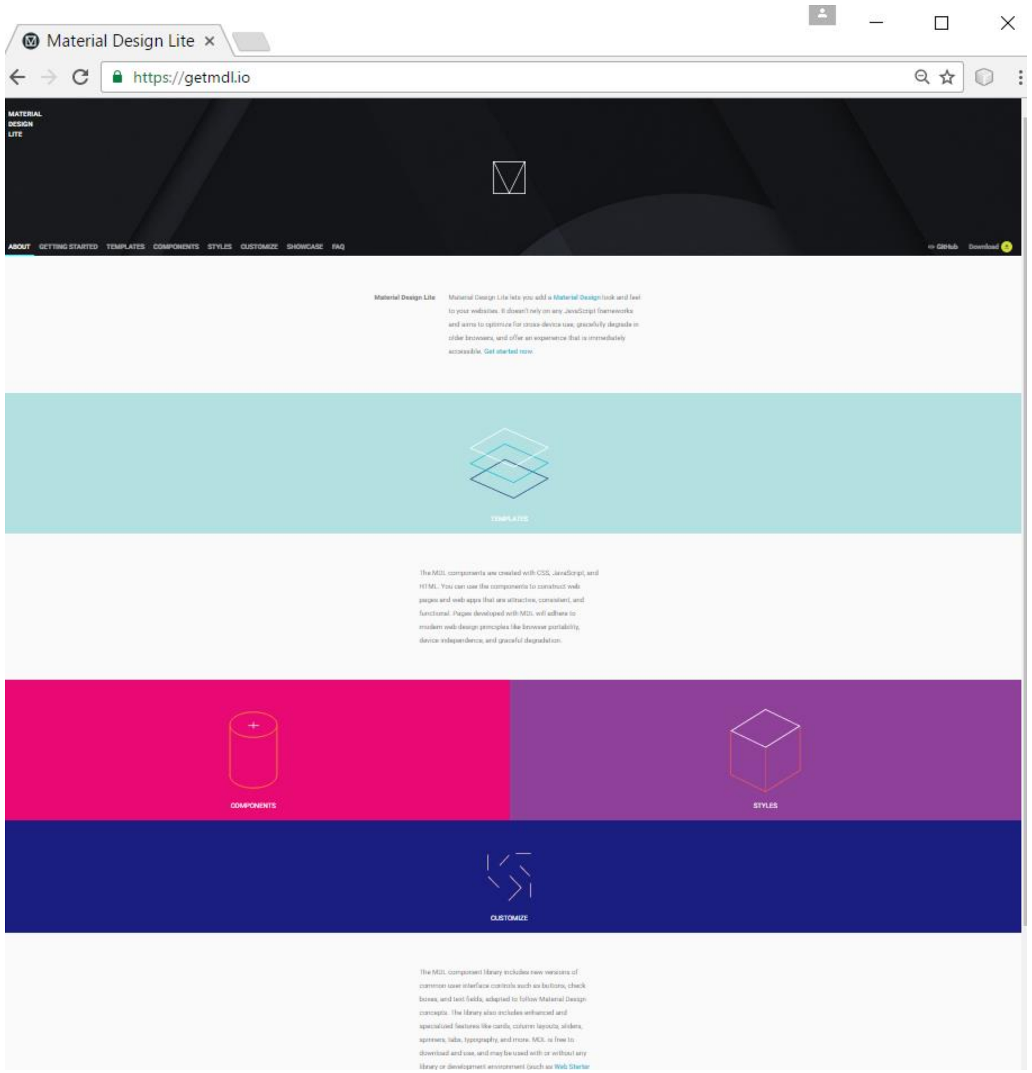
Made for everyone.

Bootstrap was made to not only look and behave great in the latest desktop browsers (as well as IE7!), but in tablet and smartphone browsers via responsive CSS as well.

Packed with features.

A 12-column responsive grid, dozens of components, javascript plugins, typography, form controls, and even a web-based Customizer to make Bootstrap your own.

Another example: Material Design Lite



Material Design Lite

Material Design Lite lets you add a Material Design look and feel to your websites. It doesn't rely on any JavaScript frameworks and aims to optimize for cross-device use, gracefully degrade in older browsers, and offer an experience that is immediately accessible. Get started now.

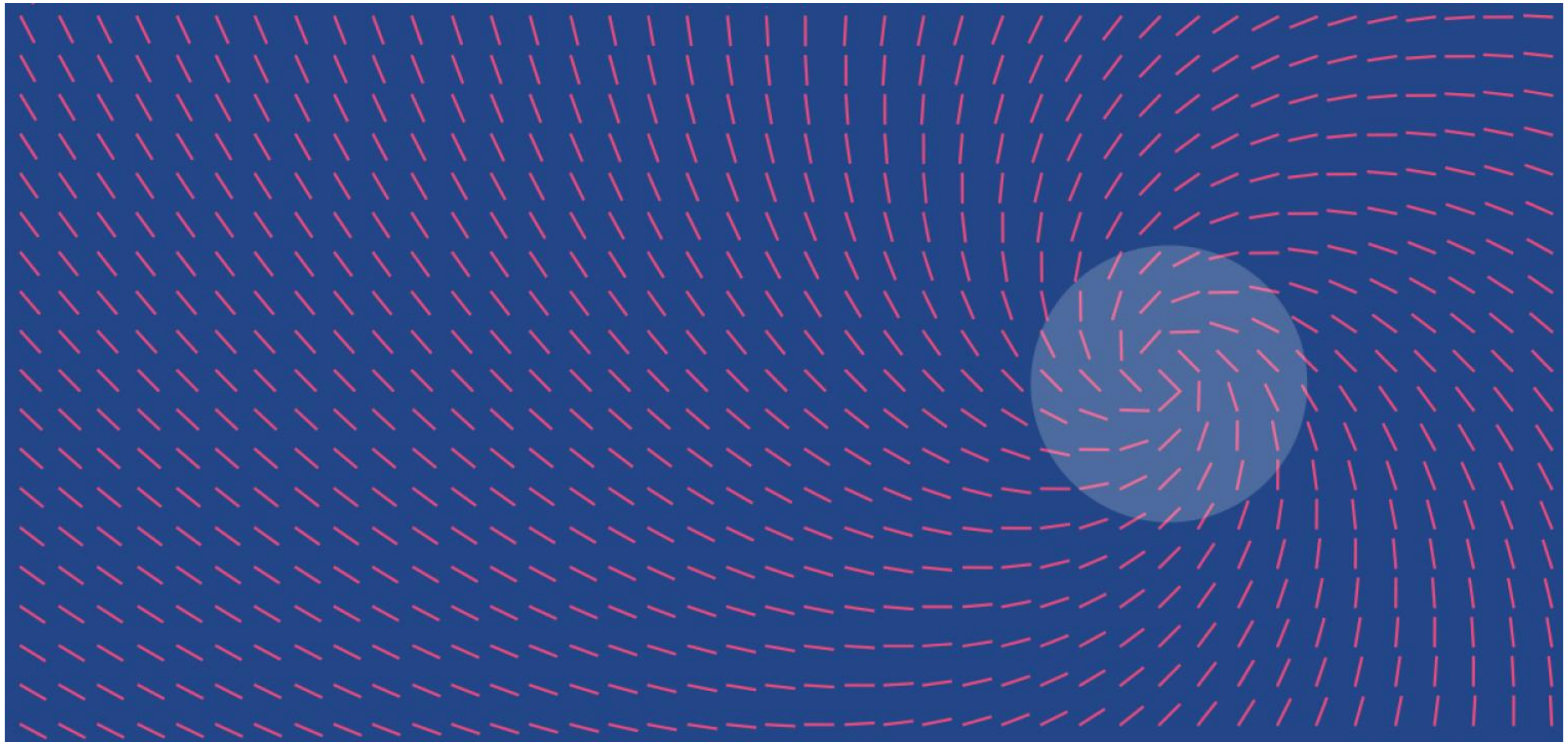
The MDL components are created with CSS, JavaScript, and HTML. You can use the components to construct web pages and web apps that are attractive, consistent, and functional. Pages developed with MDL will adhere to modern web design principles like browser portability, device independence, and graceful degradation.

The MDL component library includes new versions of common user interface controls such as buttons, check boxes, and text fields, adapted to follow Material Design concepts. The library also includes enhanced and specialized features like cards, column layouts, sliders, spinners, tabs, typography, and more. MDL is free to download and use, and may be used with or without any library or development environment (such as Web Starter

Material Design

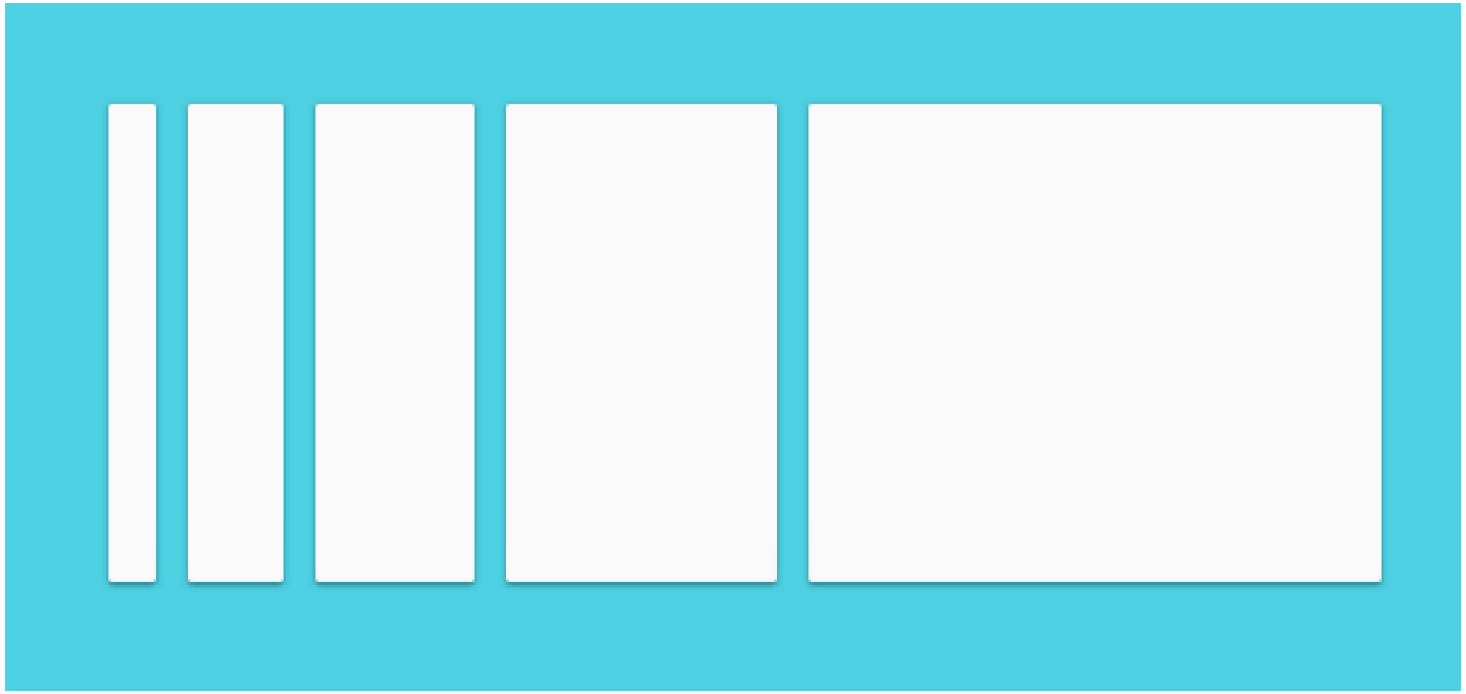
Material Design Goal #1

Create a visual language that synthesizes classic principles of good design with the innovation and possibility of technology and science



Material Design Goal #2

Develop a single underlying system that allows for a unified experience across platforms and device sizes. Mobile precepts are fundamental, but touch, voice, mouse, and keyboard are all first-class input methods.



Material is a Metaphor

UIs real enough to grab and touch