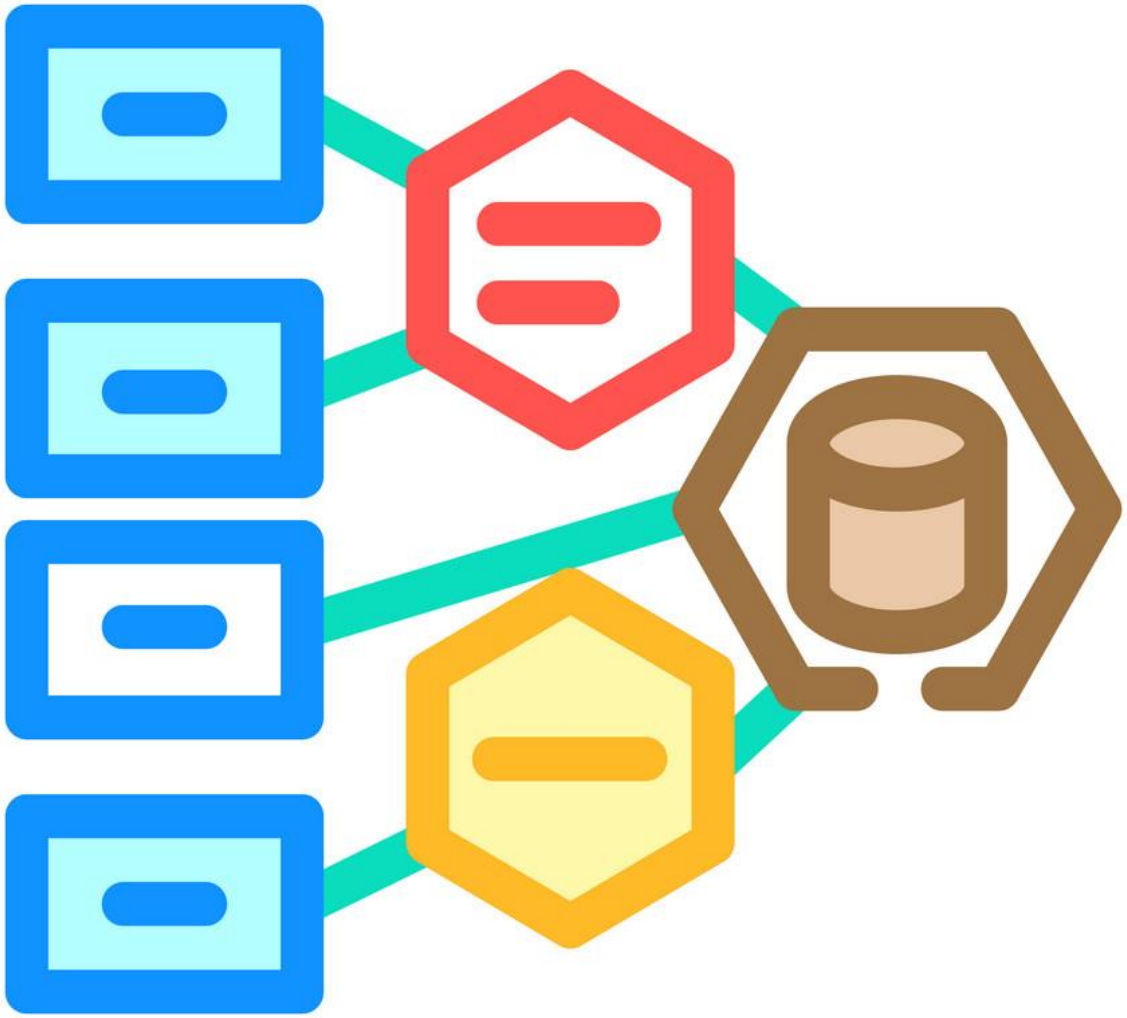
Software Design and architecture

Fall 2024

Dr. Natalia Chaudhry



User interface design

THE GOLDEN RULES

Reduce the User's Memory Load

- The interface should be designed to reduce the requirement to remember past actions, inputs, and results.
- This can be accomplished by providing visual cues that enable a user to recognize past actions, rather than having to recall them

- Establish meaningful defaults.
- The initial set of defaults should make sense for the average user, but a user should be able to specify individual preferences.
- However, a "reset" option should be available, enabling the redefinition of original default values.

- Define shortcuts that are intuitive.
- When mnemonics are used to accomplish a system function (e.g., alt-P to invoke the print function), the mnemonic should be tied to the action in a way that is easy to remember (e.g., first letter of the task to be invoked).

- Disclose information in a progressive fashion. The interface should be organized hierarchically.
- That is, information about a task, an object, or some behavior should be presented first at a high level of abstraction.
- More detail should be presented after the user indicates interest.

Make the Interface Consistent

- Allow the user to put the current task into a meaningful context.
- Many interfaces implement complex layers of interactions with dozens of screen images.
- It is important to provide indicators (e.g., window titles, graphical icons, consistent color coding) that enable the user to know the context of the work at hand.
- In addition, the user should be able to determine where he has come from and what alternatives exist for a transition to a new task.

- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.
- Once a particular interactive sequence has become a de facto standard (e.g., the use of alt-S to save a file), the user expects this in every application encountered.
- A change (e.g., using alt-S to invoke scaling) will cause confusion.

The overall process for analyzing and designing a user interface begins with the creation of different models of system function

Design Model: The designer's conceptual framework for how the system should work and look, focusing on structure, functionality, and user interaction.

User Model: The user's mental perception of how the interface works, shaped by prior knowledge, experience, and interface design cues.

Mental Model: The user's internal understanding of system behavior and expected outcomes, influencing how they predict and interact with the interface.

Implementation Model: The underlying system architecture and technical details that enable functionality, often hidden from the user.

Key Goal: Align the **design model** with the **user's mental model** for seamless interaction while ensuring the **implementation model** supports the design.

Design Model (constructed by designers):

- **Who constructs it?** UX/UI designers and system architects.
- **What does it involve?** A representation of how the interface should function, appear, and flow to meet user needs. It includes wireframes, mockups, workflows, and interaction patterns.
- **Purpose:** Communicate how the system should behave from a user perspective.

User Model (constructed by users):

- **Who constructs it?** The end-users, based on their interaction with the system.
- **What does it involve?** A personal interpretation of how the system works, developed through exploration, training, and experience.
- **Purpose:** Helps users navigate and interact with the system effectively.

Mental Model (constructed by users):	Implementation Model (constructed by developers):
•**Who constructs it?** Users, based on prior knowledge, expectations, and system cues. •**What does it involve?** A cognitive representation of how they *think* the system operates, often influenced by past experiences with similar systems. •**Purpose:** Guides users' decision-making and interaction expectations.	•**Who constructs it?** Software engineers and developers. •**What does it involve?** The actual system architecture, code, and back-end processes that enable the interface to function. •**Purpose:** Ensures the system operates as intended and supports the features envisioned in the design model.

- **Designers** bridge the gap between the user's mental model and the implementation model by creating a design model that aligns closely with user expectations.
- **Developers** translate the design model into an implementation model that faithfully supports the user experience.
- The success of an interface depends on how well the **design model** aligns with both the **mental model** and the **implementation model.**

Web application design

WEBAPP DESIGN QUALITY

Usability:

•**Definition:** How easily users can navigate and achieve their goals.

•**Example: Google Drive** - Clean UI with intuitive menus, search bar, and drag-and-drop features.

Performance:

•**Definition:** Fast response times, optimized load speeds, and scalability.

•**Example: Amazon** - Quick product searches and checkout processes, even with high traffic.

Accessibility:

•**Definition:** Inclusive design for users with disabilities (e.g., screen readers, keyboard navigation).

•**Example: BBC News** - Provides ARIA labels, alt text, and resizable text options.

Responsiveness:
•**Definition:** Seamless adaptation to different devices and screen sizes.
•**Example: Twitter** - Optimized for mobile, tablet, and desktop views.

Consistency:
•**Definition:** Uniformity in UI elements, fonts, colors, and interactions.
•**Example: Microsoft Teams** - Consistent buttons, menus, and tooltips across devices.

Security:
•**Definition:** Protection against unauthorized access and data breaches.
•**Example: PayPal** - Uses encryption, two-factor authentication, and fraud detection.

Aesthetics:
•**Definition:** Visually appealing layout with balanced colors, typography, and imagery.
•**Example: Airbnb** - Minimalist design with vibrant images and clean fonts.

Scalability:
•**Definition:** Ability to handle increased users or data without degrading performance.
•**Example: Netflix** - Supports millions of users with high-quality streaming.

Interactivity:
•**Definition:** Engaging and responsive feedback to user actions.
•**Example: Spotify** - Dynamic playlists, real-time song updates, and user-customized interfaces.

Reliability:
•**Definition:** Continuous availability and minimal downtime.
•**Example: Slack** - Reliable messaging and notification delivery even during high loads.

That’s it