G

Graphical User Interface (GUI): Components and Examples

A graphical user interface lets people interact with software through visual controls such as windows, icons, menus, buttons, and pointers.

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
4 min

CONTENTS

In this guide

01 What is a GUI?
02 Main GUI components
03 GUI versus command-line interface
04 What makes a GUI usable?
05 What makes a GUI accessible?
06 Security and privacy considerations
07 Examples
08 Sources

Quick answer: A graphical user interface, or GUI, lets people interact with a computer or application through visual objects such as windows, icons, menus, buttons, fields, and pointers. Instead of remembering every typed command, a user can see available actions

and operate them with a mouse, touch, keyboard, voice control, or assistive technology.

01 What is a GUI?

A GUI is the visible and interactive layer between a person and software. It represents information and actions with graphical components while the operating system or application performs the underlying work. Desktop operating systems, phone home screens, web applications, ATMs, car displays, and smart televisions all use graphical interfaces.

“Graphical” does not mean mouse-only. A well-built GUI exposes the same controls to keyboard users and assistive technologies. Visual presentation, programmatic names, states, focus, and feedback must remain coordinated.

02 Main GUI components

- **Windows and panels** group content or tasks and may be moved, resized, minimized, or arranged.
- **Icons** visually represent applications, files, actions, status, or navigation destinations.
- **Menus and toolbars** organize available commands and commonly used actions.
- **Controls** include buttons, links, checkboxes, switches, lists, sliders, tabs, and text fields.
- **Pointer and focus** indicate where mouse, touch, or keyboard input will act.
- **Feedback** communicates progress, success, errors, selection, and disabled or active states.

The classic shorthand WIMP refers to windows, icons, menus, and pointer, but modern GUIs also use gestures, search, notifications, voice input, responsive layouts, and direct manipulation.

03 GUI versus command-line interface

A command-line interface (CLI) accepts typed commands and displays text output. A GUI emphasizes discoverability and visible feedback, while a CLI can be faster for repeatable, remote, or automated work. Neither is automatically more powerful or secure. Many systems offer both: a visual control panel for common tasks and a shell or API for precise administration.

A GUI can reduce syntax mistakes but may hide important details or make bulk changes slow. A CLI can reproduce a sequence exactly but a pasted command may cause broad changes immediately. Users should understand the scope and privilege of either action before confirming it.

04 What makes a GUI usable?

A usable interface gives controls clear labels, follows familiar patterns, shows system status, prevents avoidable errors, and offers a safe way to undo or recover. Related items should be grouped, destructive actions distinguished from routine actions, and important choices explained in the user's language. Consistency reduces the need to relearn behavior on each screen.

05 What makes a GUI accessible?

1. Every function can be reached and operated with a keyboard, and focus never becomes trapped.
2. Focus order is meaningful and the current focus indicator is visible.
3. Text and controls have sufficient contrast; color alone is not used to convey status.
4. Controls have programmatic names, roles, values, and states for screen readers.
5. Forms use persistent labels and useful error messages rather than placeholder text alone.
6. The layout adapts to zoom, larger text, touch input, and different viewport sizes.

Accessibility cannot be confirmed by an automated scan alone. Keyboard testing, screen-reader testing, zoom and reflow checks, and evaluation by people with different access needs reveal interaction problems that markup checks miss.

06 Security and privacy considerations

Attackers imitate familiar GUIs in phishing pages, fake update dialogs, and permission prompts. A polished interface or padlock is not proof that a request is legitimate. Verify the application, site domain, publisher, and requested permission. Developers should avoid dark patterns, protect

sensitive values from unnecessary display, prevent clickjacking where relevant, and require clear confirmation for consequential actions.

07 Examples

File managers use folders, lists, breadcrumbs, and drag-and-drop. Browsers combine an address field, tabs, navigation controls, page content, and permissions. Mobile interfaces replace a permanent pointer with touch targets and gestures. A screen reader can present the same GUI as a structured sequence of headings, controls, names, and states when the interface exposes correct semantics.

08 Sources

The interaction and accessibility guidance follows the W3C Web Accessibility Initiative's accessibility principles and interface design guidance.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com