INFORMATION SYSTEMS

UNIT CODE: ICT/CU/CS/CR/06/6/A

Relationship to Occupational Standards

This unit addresses the unit of competency: Develop an Information System

Duration of Unit:150 hours

Unit Description

This unit covers the competencies required to develop an information system. It involves understanding fundamentals of information systems, understanding the software development process, demonstrating human computer interaction principles, understanding the VB.net programming environment and developing and testing a VB.NET application.

Summary of Learning Outcomes

1. Understand fundamentals of Information Systems
2. Understand the Software Development Process
3. Demonstrate Human Computer Interaction Principles
4. Understand the VB.NET programming environment
5. Develop and test a VB.NET application

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Understand fundamentals of Information Systems	• Information systems ✓ Definition ✓ Components • Types of information systems ✓ Transaction Processing Systems ✓ Management Information Systems ✓ Decision Support Systems ✓ Executive Information Systems ✓ Office Automation Systems • Emerging trends in information systems • Recommendation of information systems for different scenarios	• Oral questioning • Written tests • Practical tests

	• Information system security ✓ Definition ✓ Information security management system ✓ Tools for information system security ✓ Firewalls ✓ Virtual private networks • Mobile security ✓ Geolocation software ✓ Remote data removal software • Web security ✓ Cyber security ✓ Technologies ✓ Web threats ✓ Defence strategies	
2. Understand the Software Development Process	• Software Development Life Cycle • Software Development Methodologies ✓ Waterfall ✓ Spiral ✓ Rapid Application Development ✓ Agile Development • Modeling techniques ✓ Data Flow Diagrams ✓ Entity Relation Diagrams ✓ UML diagrams • Creation of models for given scenarios	• Written tests • Oral questioning • Practical tests
3. Demonstrate Human Computer Interaction Principles	• Human Computer Interaction ✓ Definition ✓ Role of interaction design ✓ Interaction styles ✓ Interaction elements ✓ Mistakes in interaction design • Interface design principles • Prescribing interaction choices and recognition of interaction flaws	• Practical • Oral questioning • Observation • Written tests

4. Understand the VB.NET programming environment	• The .Net framework ✓ Applications supported ✓ Components of the .Net framework • Installation of Visual Studio • Features of VB.Net • The Integrated Development Environment (IDE) ✓ Definition of IDE ✓ Parts of VB.Net IDE • VB.Net program structure ✓ VB.NET syntax ✓ Namespace declaration ✓ Class or module ✓ Procedures ✓ Data types, variables, constants ✓ The Main procedure ✓ Statements and Expressions (Variable declarations, operations, control statements) ✓ Comments • Creating aVB.Net project ✓ Saving Forms and Project ✓ Compiling a Project	• Practical tests • Oral tests • Written tests
5. Develop and test a VB.NET application	• Basic VB.Net Controls ✓ Controls and their purpose ✓ Standard naming conventions for controls • Elements of a control ✓ Properties ✓ Methods ✓ Events • Demonstrating Properties, Methods and Events ✓ Properties for basic controls ✓ Setting properties at design time and run time ✓ Methods for basic controls ✓ Events for basic controls • Demonstrating event handling	• Practical tests • Oral tests • Written tests

	✓ Mouse events ✓ Keyboard events • Designing VB.NET form using HCI principles • Connection of VB.Net applications to a database ✓ ADO.Net object model ✓ Demonstrating Database connection using the Data Provider ✓ Demonstrating creation of tables using Dataset components • Deployment of VB.NET VB.Net applications ✓ Purpose deployment ✓ Demonstrating deployment steps	

Suggested Methods of Instruction

- Presentations and practical demonstrations by trainer;
- Guided learner activities and research to develop underpinning knowledge;
- Supervised practical assignments and projects;
- Visiting expert from the ICT sector;
- Industrial visits

Recommended Resources
Tools

- Visual Studio, CASE software, UX/UI software

Equipment

- Computer

Materials and supplies

- Instructional materials
- Stationery

Reference materials

- Trainer-recommended resources including web resources
- Visual Studio Documentation

1. Understand fundamentals of Information Systems

Introduction to information systems

Information system, an integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products.

Business firms and other organizations rely on information systems to carry out and manage their operations, interact with their customers and suppliers, and compete in the marketplace. Information systems are used to run inter-organizational supply chains and electronic markets. For instance, corporations use information systems to process financial accounts, to manage their human resources, and to reach their potential customers with online promotions.

What Are The Types Of Information Systems?

1. Knowledge Work System

There are different knowledge management systems that an organization implements to ensure a continuous flow of new and updated knowledge into the company and its processes. A knowledge work system (KWS) is one of the knowledge management systems that ease the integration of new information or knowledge into the business process.

Furthermore, KWS also offers support and resources to various knowledge creation techniques, artificial intelligence applications, and group collaboration systems for knowledge sharing, among others. It also uses graphics, visuals, etc., to disseminate new information. Below are some of the applications that work on the core fundamentals of KWS.

Designers often use computer-aided design systems (CAD) to automate their design process.

Financial workstations are used to analyze huge amounts of financial data with the help of new technologies.

Virtual reality systems are found in the scientific, education, and business fields for using graphics and different systems to present data.

2. Management Information System

The management information system provides aid to managers by automating different processes that were initially done manually. Business activities like business performance tracking and analysis, making business decisions, making a business plan, and defining workflow. It also provides feedback to the managers by analyzing the roles and responsibilities.

A management information system is considered a significant application that helps managers immensely. Here are some of the advantages of the information system:

- It enhances the efficiency and productivity of the company
- It provides a clear picture of the organization’s performance
- It adds value to the existing products, introduces innovation and improves product development
- It assists in communication and planning for business processes
- It helps the organization provide a competitive advantage

3. Decision Support System

A decision support system is an information system that analyses business data and other information related to the enterprise to offer automation in decision-making or problem-solving. A manager uses it in times of adversities arising during the operation of the business. Generally, the decision support system is used to collect information regarding revenue, sales figures or inventory. It is used across different industries, and the decision support system is a popular information system.

4. Office Automation System

An office automation system is an information system that automates different administrative processes like documenting, recording data, and office transactions, among others. The office automation system is divided into managerial and clerical activities. Here are some of the business activities that are done under this type of information system:

- Email
- Voice mail

- Word processing

5. Transaction Processing System

The transaction processing system automates the transaction collection, modification, and retrieval process. The peculiar characteristic of this type of information system is that it increases the performance, reliability and consistency of business transactions. It helps businesses perform daily operations smoothly without hassle.

Once you are well-versed with different types of information systems, understanding the application of these systems becomes easy to comprehend. Therefore, in the last part of the article, we will look into applying information systems.

6. Executive Support System

An Executive Support System or ESS helps top-level executives to plan and control workflow and make business decisions. It is very similar to Management Information System or MIS.

Here are some of the unique characteristics of ESS:

It provides great telecommunication, better computing capabilities, and effective display options to executives.

It enables them with information through static reports, graphs, and textual information on demand.

It helps monitor performances, track competitors' strategies, and forecast future trends, among **others.**

How to Apply Information Systems in Business?

Here are some of the business activities that require the intervention of an information system.

Enterprise resource planning (ERP)

Applying information systems to enterprise resource planning helps automate business administration and planning functions.

Supply chain management (SCM)

Information systems provide a common forum to connect with different parties in supply chain management. Moreover, it makes communication between parties easy and resourceful.

Customer relationship management (CRM)

Many information systems help in realizing customer requirements. Furthermore, other information applications help companies interact with their audience easily and hassle-free.

To understand the use of information technology and its ancillary systems, many business professionals pursue different courses. Emeritus India offers some of the best IT courses in partnership with various Indian and international institutes. So, enroll in our famous IT courses to enhance your career. To use technical knowledge and innovation, explore the Certificate Programme in Digital Transformation and Innovation by IIMI (Indore)

TASK

Visit different section of OTTI with a view of identifying the following aspects

1. **Types of information systems in place?**
2. **Importance that information systems in place**
3. **Short comings of such a information system**

2. Understand the Software Development Process

What is SDLC?

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.

Why is SDLC important?

Software development can be challenging to manage due to changing requirements, technology upgrades, and cross-functional collaboration. The software development lifecycle (SDLC) methodology provides a systematic management framework with specific deliverables at every stage of the software development process. As a result, all stakeholders agree on software development goals and requirements upfront and also have a plan to achieve those goals.

Here are some benefits of SDLC:

Increased visibility of the development process for all stakeholders involved

Efficient estimation, planning, and scheduling

Improved risk management and cost estimation

Systematic software delivery and better customer satisfaction

How does SDLC work?

The software development lifecycle (SDLC) outlines several tasks required to build a software application. The development process goes through several stages as developers add new features and fix bugs in the software.

The details of the SDLC process vary for different teams. However, we outline some common SDLC phases below.

1. **Plan**

The planning phase typically includes tasks like cost-benefit analysis, scheduling, resource estimation, and allocation. The development team collects requirements from several stakeholders such as customers, internal and external experts, and managers to create a software requirement specification document.

The document sets expectations and defines common goals that aid in project planning. The team estimates costs, creates a schedule, and has a detailed plan to achieve their goals.

2. **Design**

In the design phase, software engineers analyze requirements and identify the best solutions to create the software. For example, they may consider integrating pre-existing modules, make technology choices, and identify development tools. They will look at how to best integrate the new software into any existing IT infrastructure the organization may have.

3. **Implement**

In the implementation phase, the development team codes the product. They analyze the requirements to identify smaller coding tasks they can do daily to achieve the final result.

4. **Test**

The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

5. **Deploy**

When teams develop software, they code and test on a different copy of the software than the one that the users have access to. The software that customers use is called production, while other copies are said to be in the build environment, or testing environment.

Having separate build and production environments ensures that customers can continue to use the software even while it is being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environment configuration, and installation.

6. **Maintain**

In the maintenance phase, among other tasks, the team fixes bugs, resolves customer issues, and manages software changes. In addition, the team monitors overall system performance, security, and user experience to identify new ways to improve the existing software.

What are SDLC models?

A software development lifecycle (SDLC) model conceptually presents SDLC in an organized fashion to help organizations implement it. Different models arrange the SDLC phases in varying chronological order to optimize the development cycle. We look at some popular SDLC models below.

1. Waterfall

The waterfall model arranges all the phases sequentially so that each new phase depends on the outcome of the previous phase. Conceptually, the design flows from one phase down to the next, like that of a waterfall.

Pros and cons

The waterfall model provides discipline to project management and gives a tangible output at the end of each phase. However, there is little room for change once a phase is considered complete, as changes can affect the software's delivery time, cost, and quality. Therefore, the model is most suitable for small software development projects, where tasks are easy to arrange and manage and requirements can be pre-defined accurately.

2. Iterative

The iterative process suggests that teams begin software development with a small subset of requirements. Then, they iteratively enhance versions over time until the complete software is ready for production. The team produces a new software version at the end of each iteration.

Pros and cons

It’s easy to identify and manage risks, as requirements can change between iterations. However, repeated cycles could lead to scope change and underestimation of resources.

3. Spiral

The spiral model combines the iterative model's small repeated cycles with the waterfall model's linear sequential flow to prioritize risk analysis. You can use the spiral model to ensure software's gradual release and improvement by building prototypes at each phase.

Pros and cons

The spiral model is suitable for large and complex projects that require frequent changes. However, it can be expensive for smaller projects with a limited scope.

4. Agile

The agile model arranges the SDLC phases into several development cycles. The team iterates through the phases rapidly, delivering only small, incremental software changes in each cycle. They continuously evaluate requirements, plans, and results so that they can respond quickly to change. The agile model is both iterative and incremental, making it more efficient than other process models.

Pros and cons

Rapid development cycles help teams identify and address issues in complex projects early on and before they become significant problems. They can also engage customers and stakeholders to obtain feedback throughout the project lifecycle. However, overreliance on customer feedback could lead to excessive scope changes or end the project midway.

3.Demonstrate Human Computer Interaction Principles

What is human-computer interaction (HCI)?

Human-computer interaction (HCI) is a field of study that focuses on the interaction between a human (the person using the system) and a computer system (the machine or network of machines that run the system).

Between the human and computer is a user interface that we design to facilitate smooth interactions. A primary focus of HCI is to help us create systems that make these interactions feel more like a natural conversation between people.

In the 1980s when HCI emerged, it focused on making desktop computers easier to use. But as technology evolved beyond desktops to include mobile phones, smartwatches, and other devices, its focus broadened to encompass a wider range of user experiences and interaction methods.

Goals and objectives of HCI

HCI aims to bridge the gap between humans and technology. Some of its goals and objectives include:

- **Improve usability** — Ensuring that interfaces are easy for our users to learn and use, reducing the effort required to perform tasks
- **Enhance accessibility** — Designing interfaces that all users can access and use easily, regardless of their abilities
- **Increase efficiency** — Helping users complete tasks more quickly and accurately by minimizing the number of steps needed to complete a task and optimizing the interface's functionalities
- **Promote user-centered design** — Ensuring that digital products meet our users' needs, preferences, and limitations, potentially by involving the users throughout the design process
- **Improve user experience** — Creating positive user experiences, ensuring that their interactions with technology are effective, engaging, and enjoyable

Ultimately, we want to make human-computer interactions enjoyable, not just functional. So, HCI looks at the user's emotional experience as well as practical considerations to help us create more engaging, intuitive, and aesthetically pleasing interfaces.

Types of interaction in HCI

Now let's look at six ways humans can interact with a computer system:

1. **Graphical user interfaces (GUIs)** — The most common form of interaction for most computer users. GUIs allow users to interact with their devices using visual components. On an ecommerce platform, for instance, you can scan through images to choose the products you want to buy
2. **Touch interfaces** — Allow users to interact with the system through touch-based gestures such as swiping, tapping, and pinching. An example is the pinch-to-zoom feature on a smartphone

3. **Command-line interfaces (CLIs)** — Use typed commands to interact with the system. An example is the built-in terminal application on most operating systems (Windows, Mac, Linux) with a text-based interface, where you can type commands to interact with the system
4. **Voice user interfaces (VUIs)** — Allow users to interact with a system through voice commands. Examples include Siri and Alexa. These are particularly useful to users with mobility impairments and those in a hands-free environment.
5. **Natural user interfaces (NUIs)** — Use gestures and body movements to make interactions with a system more natural and intuitive. Examples include virtual reality gaming and Microsoft Kinect
6. **Multimodal interfaces** — Combine multiple modes of interaction, such as vision, touch, and voice, to provide a more flexible user experience. An example is when you search for an item with a voice command and select it with touch

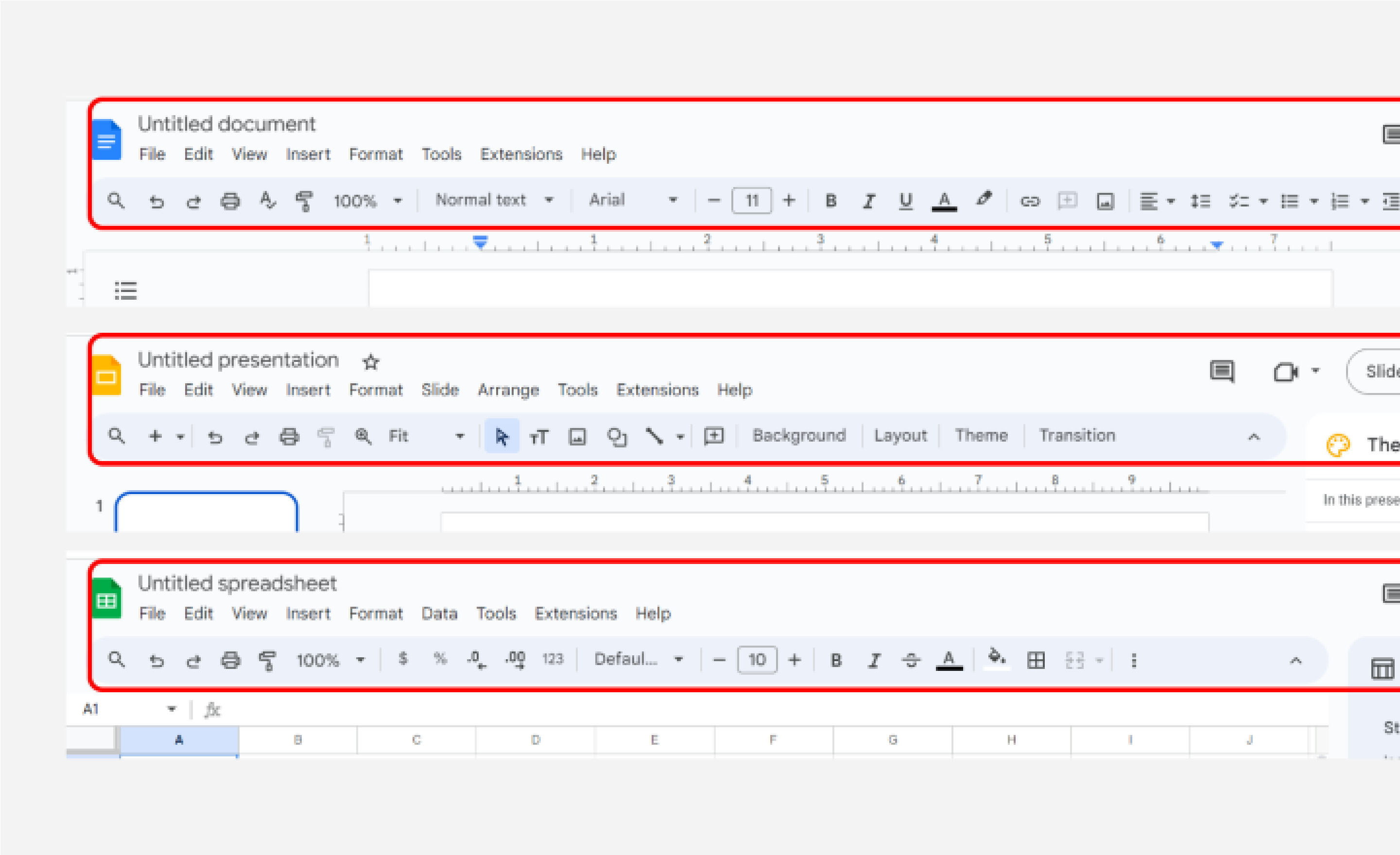
These interface types offer different ways for our users to interact with our systems, but it's also important to ensure these interactions are seamless and intuitive. Let's explore nine principles to help us do so.

9 principles of HCI

The following nine principles of HCI can help guide you in creating more enjoyable and user-friendly interfaces.

1. Consistency

Ensure that similar actions and elements are represented the same way throughout the interface so that users won't need to learn new interactions for every task. Also, reuse components and behaviors to make it easy for users to transfer knowledge from one task to another:


Untitled document
File Edit View Insert Format Tools Extensions Help
100% Normal text Arial 11
Untitled presentation
File Edit View Insert Format Slide Arrange Tools Extensions Help
Fit Background Layout Theme Transition
Slide
The
In this prese
Untitled spreadsheet
File Edit View Insert Format Data Tools Extensions Help
100% $ % .0 .00 123 Defaul... 10
A1 fx
A B C D E F G H I J

Screenshots from Google Docs, Google Slides, and Google Sheets.

A great example of this principle is the consistent interface across the three Google Workspace tools shown above. Although the toolbars are contextual based on the product, most elements are in the same location across all interfaces.

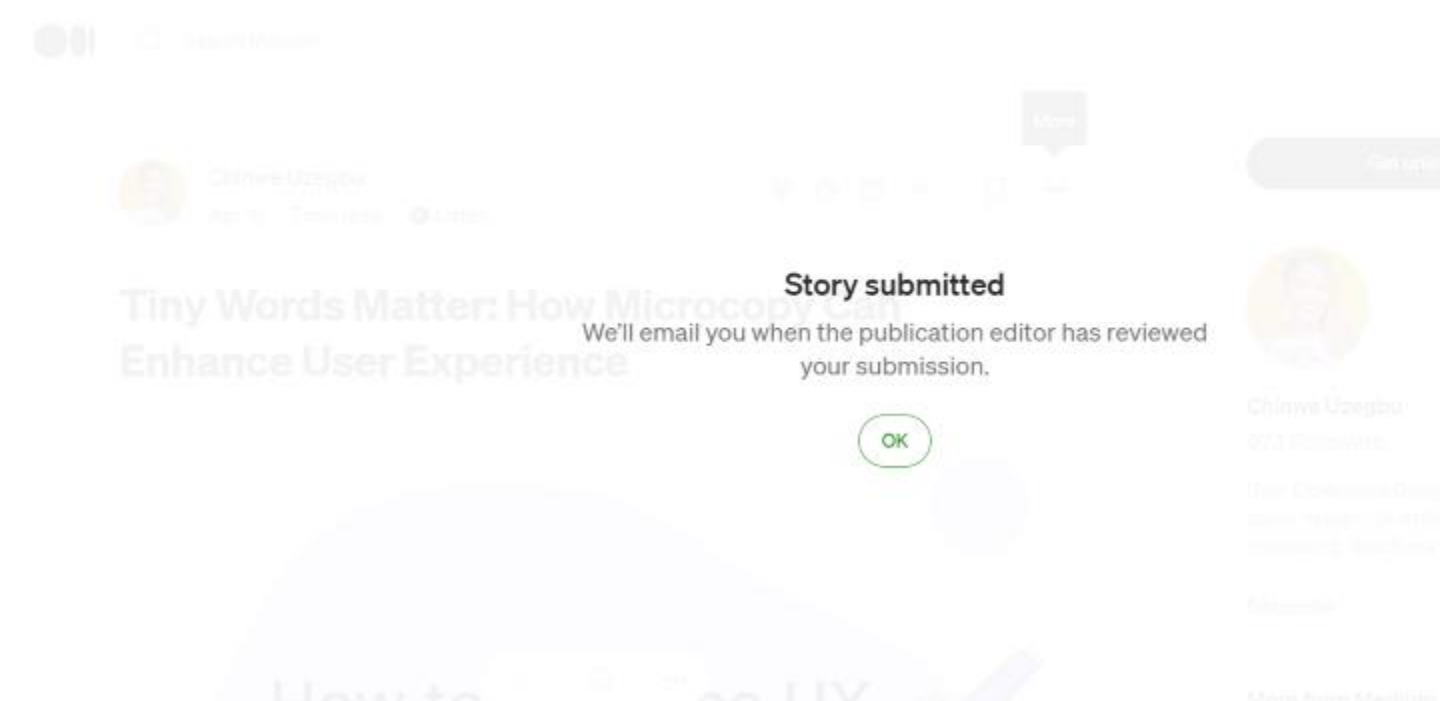
Jason Fill
@JasonFill · Follow

I have to say @LogRocket has to be one of the best tools our team has ever adopted. Based on the data (videos) we are seeing, it is allowing us to refine the product in ways we never would have imagined. Great product, highly recommend for any SaaS product.

1:28 PM · Jan 15, 2018

2. Feedback

Let users know the results of their actions. This helps users understand whether their actions were successful and helps them use the system correctly:



Story submitted

We'll email you when the publication editor has reviewed your submission.

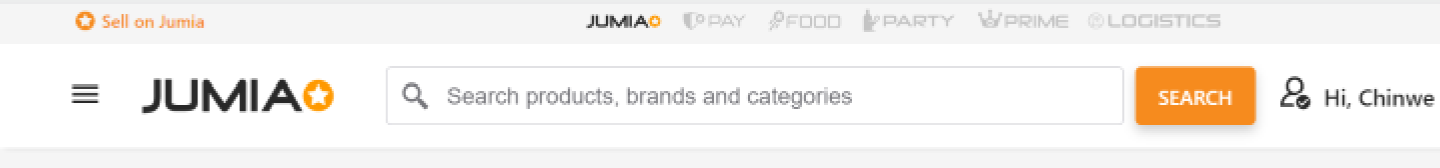
Tiny Words Matter: How Microcopy Can Enhance User Experience
OK

Screenshot from Medium

For example, this confirmation message from Medium lets a user know that their story has been successfully submitted and what next step to expect.

3. Visibility

Users should be able to easily see and understand the state of the system and its components so they can take the right actions:

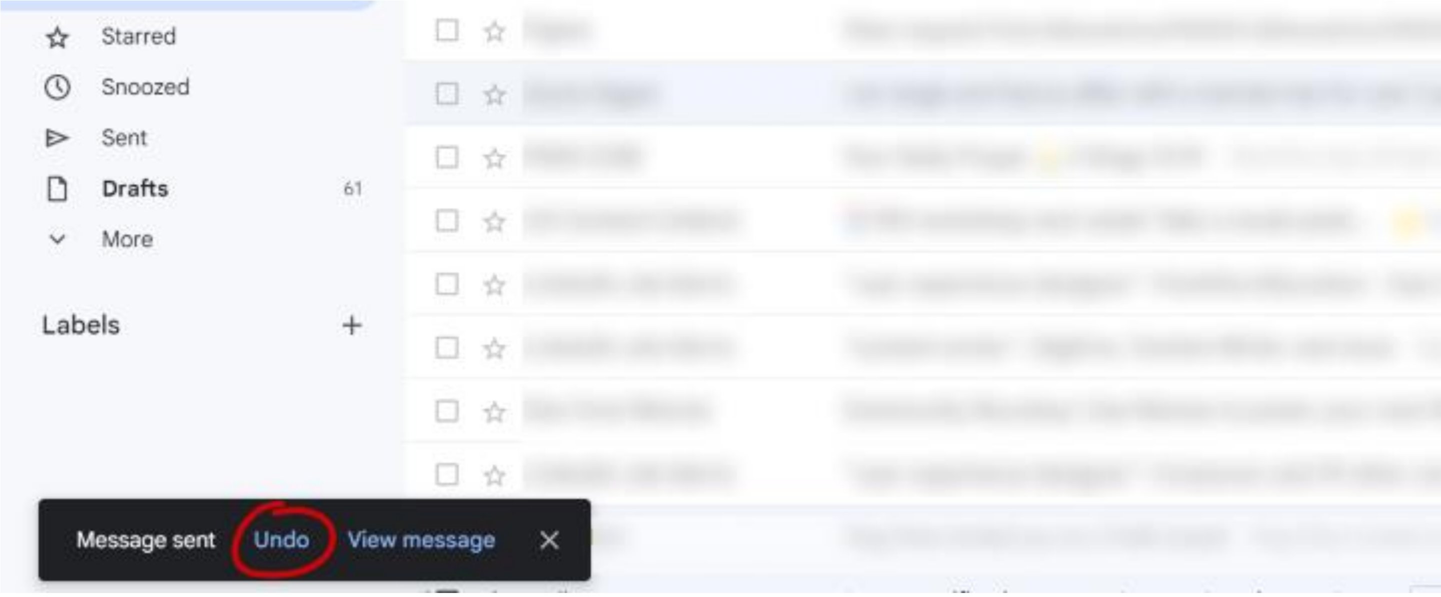


Screenshot from Jumia

An example of this is an always-visible icon showing the number of items in a cart, as shown in the screenshot from the Jumia website above. This tells users the current status of their cart without navigating away from their task — more shopping.

4. Error prevention and recovery

Make it easy for users to undo their last action and recover from mistakes without significant consequences:



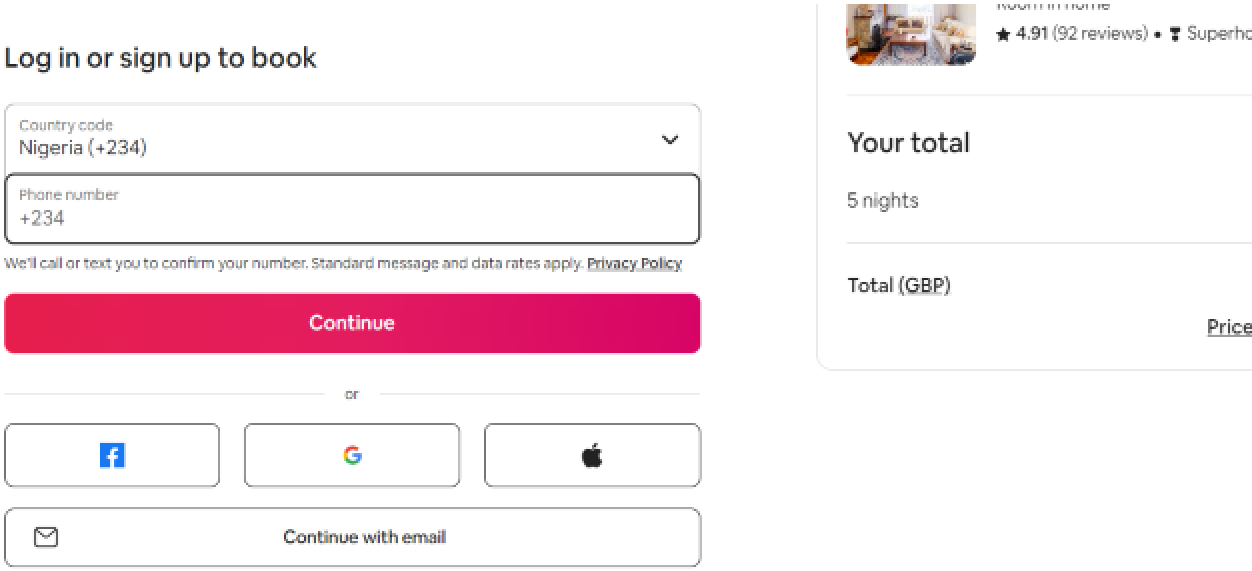
Message sent

Screenshot from Gmail

An example of this principle is the “Undo” option that appears for a few seconds after a user sends an email on Gmail. This allows the user to retract an email if it was sent by mistake.

5. Constraints

Limit the range of possible actions in an interface to avoid unintended actions or invalid selections. For example, if you had a form with a phone number field, you could ensure the user can only input numbers, as shown in this example from Airbnb:



Log in or sign up to book

Country code: Nigeria (+234)

Phone number: +234

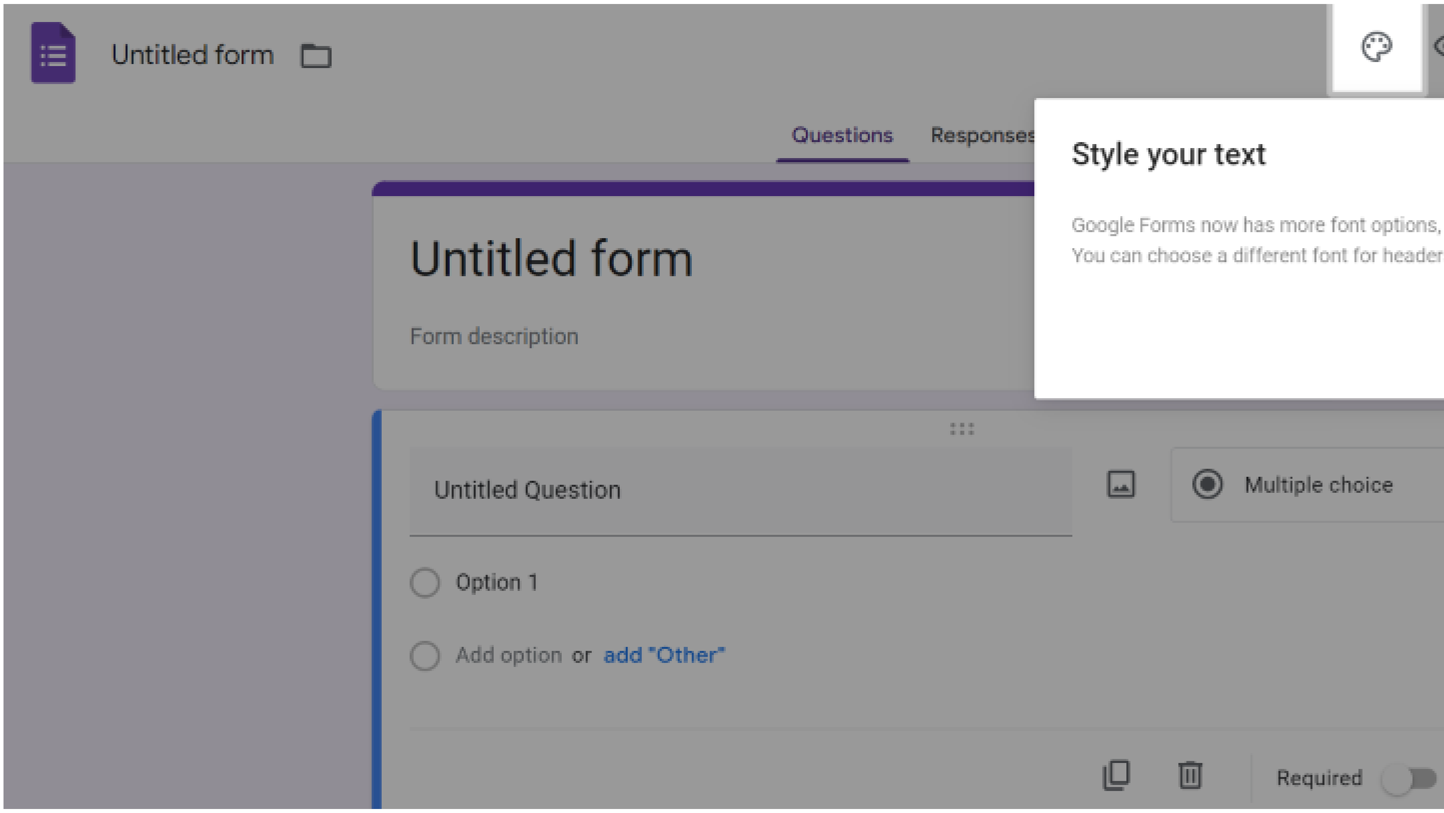
We'll call or text you to confirm your number. Standard message and data rates apply. Privacy Policy

Screenshot from Airbnb

But be careful when implementing constraints — too many can make an interface feel restrictive.

6. Learnability

Provide users with brief tutorials and guided tours to show them how to navigate and use the product's features quickly. A simple example of this principle is the tooltips in Google Forms that help new users understand how to use the different tools in the interface:



Style your text

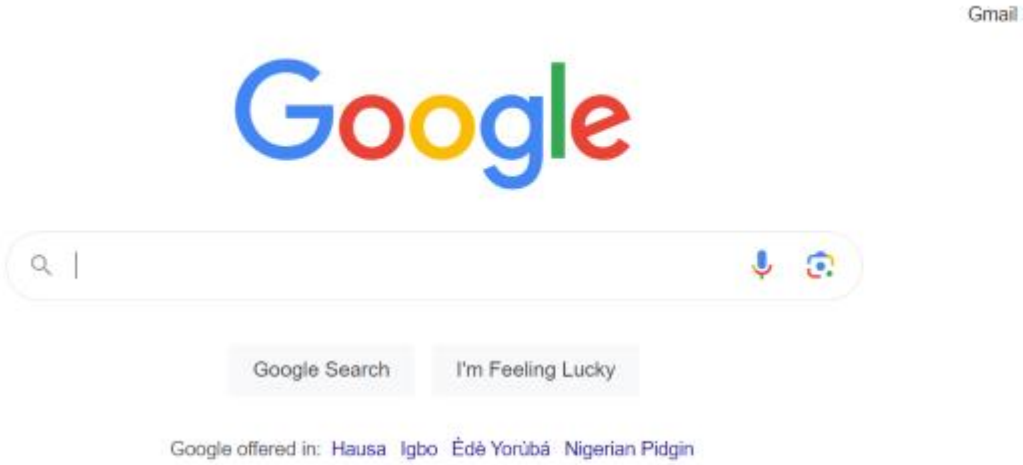
Google Forms now has more font options,
You can choose a different font for header

Screenshot from Google Forms

Onboarding tutorials can save users the extra time they might spend figuring things out on their own.

7. Simplicity

Use minimalistic design to ensure users can perform their primary tasks without distractions. As you can see in the example below, Google's search engine homepage only has a search bar and a few buttons, leaving almost no room for confusion:



Screenshot from Google

Minimizing the number of elements with high visual priority on a page allows you to guide users to the most important actions.

8. Mental models

Design interfaces to align with users' real-world experiences. This makes it easy for them to understand how things work based on their existing knowledge.

You can see this principle being applied in the use of icons that resemble objects or concepts from the real world. An example is using a trash can icon for deleting files or a magnifying glass icon for search functions.

9. Affordance

Use intuitive features to make clear to users how to interact with interface elements. Affordances are important as they let users know what actions are possible in an interface:

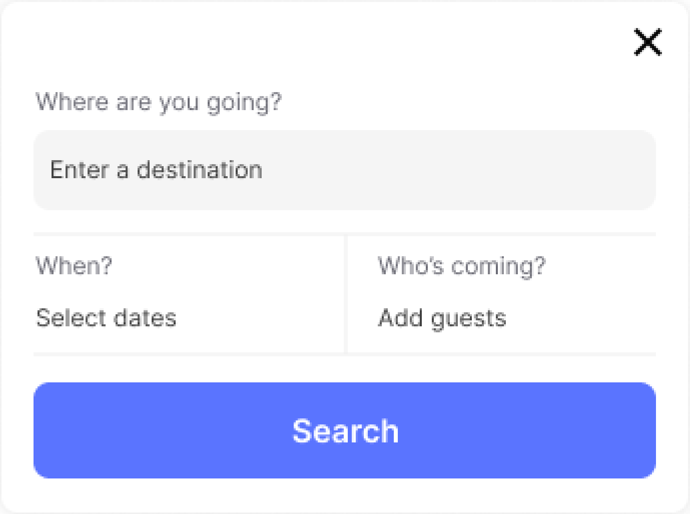


Image by the author

An example is using a button with the word “Search” to indicate that users can trigger a search by pressing it. If the example above looks familiar to you, you’re probably remembering my in-depth exploration of affordances in UX design — check it out if you’d like to learn more about this principle.

Use visual cues — such as the button’s shape, color, and style — that align with the user’s prior experiences, making it easy for them to understand the button’s purpose.

Interaction design in HCI

Interaction design (IxD), as the name suggests, focuses on designing the interaction between users and products. Considering the users’ needs and abilities will help you create intuitive actions (clicks, taps, voice commands) and clear system responses (visual cues and sounds) for a smooth UX.

IxD is a central component of HCI. It directly influences how users perceive and interact with a system and ensures that the user experience is smooth and efficient, from making navigation easier to guiding users and preventing errors.

Now, let’s look at some practical applications of HCI in some industries and their impact on user experience.

Applications of HCI

You can leverage HCI’s types and principles across various industries to improve user experiences and boost productivity. Here are some specific examples of how HCI is integrated into different sectors:

1. **Microsoft Office Suite (Software development)** — Microsoft has constantly enhanced its user interface based on HCI principles. In 2007, they introduced the Ribbon interface to improve user experience by enabling users to find, understand, and use commands with minimal clicks
2. **Roomba vacuum (Robotics)** — Roomba, one of the most popular consumer robotic vacuum cleaners, makes it easy for users to interact with it using simple commands. Its user-friendly interface and ability to guide itself while cleaning make it accessible to a wide audience
3. **Learning management systems (Education)** — Online learning platforms like Duolingo use HCI principles to create intuitive, engaging features like interactive quizzes, assessments, and progress-tracking dashboards that make the platform more engaging and improve learning outcomes
4. **Electronic health record systems (Healthcare)** — EHR systems such as NextGen and Cerner use HCI principles to build easy-to-use UIs where healthcare providers can enter, retrieve, and analyze patient medical records. This improves workflow, reduces errors, and ultimately enhances patient care

These examples illustrate how to take the principles of HCI beyond theory and make a profound impact on improving user experiences.

Final words

As we increasingly turn to technology to get things done