

Human Computer Interaction (In English)

Prof. Rajiv Ratn Shah

**Department of Computer Science and Engineering
Institute IIT Madras**

Inclusivity, Accessibility and Design Principles

Good morning, everyone. I am Ritvik Bamba, and I am your TA for the course on HCI. For today's tutorial, we will be covering inclusivity, accessibility, and design principles. First, we need to discuss accessibility and inclusiveness. Accessibility refers to making technology usable for people with disabilities. It's about ensuring that individuals with different physical or cognitive challenges like vision, hearing, or mobility impairments can easily use websites, apps, or devices.

Meanwhile, inclusiveness is broader and aims to make technology usable for everyone, regardless of their background, abilities, or situation. It considers diversity in culture, language, age, socioeconomic status, and physical or cognitive abilities. Inclusive design ensures that a product doesn't exclude any group of people. Sometimes these terms may be used interchangeably, which is somewhat wrong because they actually mean a little different.

Inclusiveness focuses on making the experience welcoming and usable by people from diverse backgrounds. It addresses a wider range of needs, including cultural, socioeconomic, and personal diversity, whereas accessibility focuses on removing barriers for users with disabilities to make the system usable. It addresses specific disabilities including visual, auditory, motor and cognitive impairments. Next we move on to some examples of the accessible design. As you can see in the first example, here we can see that YouTube also gives transcripts which helps the users with disabilities to listen to the videos with the help of timestamps as well as the captions also some games offer colorblind modes which help people with color deficiencies to distinguish between game elements as you can see here you can turn the colorblind mode on and off and also set its strength as per your suitability As we move on to some more examples, some interfaces offer keyboard-only navigation, which are fully navigable by a keyboard rather than relying on a mouse, which aids users with limited motor abilities, allowing them to access all the functionality using just the keyboard shortcuts.

Some softwares also offer color contrast options. As you can see in the picture here, all the images are high contrast themes and the customizable color settings makes the digital

content accessible for users with visual impairments to ensure readability and clarity. Amazon's Alexa for seniors is another one such example. Amazon Echo devices have a senior living feature that allows seniors and their caregivers to set reminders, communicate with family, and access emergency assistance hands-free, thus improving the user experience overall. Gmail is also one such example of accessible design, where it is optimized for screen readers, providing visually impaired users with keyboard shortcuts as well as voice navigation.

This makes Gmail more accessible for users with vision impairments, allowing them to independently manage their own communications. Now we move on to some examples of inclusive design. The first we have is Twitter. It allows users to find the languages they use and allows them to choose the languages they would like to read in. This makes Twitter inclusive for everyone irrespective of their linguistic backgrounds.

Uber is also one such example where it gives visual flashing notification for new ride requests and options to communicate with the passenger via text. This improves inclusivity on both the driver as well as the passenger side. The assistive touch feature on the iPhone is a feature that allows individuals with limited motor functions to allow them to use gestures, tabs, and customizable on-screen menus instead of the physical buttons. This improves inclusivity and ensures a seamless user experience throughout. As you can see in the second picture, the first image is actually a chair that just fits everyone, while the second picture is a personally adjustable chair.

The office space with adjustable desks and chairs makes up a good example for an inclusive design. The next we have is Nike. Nike's adaptive clothing like flyers, shoes, and apparel designed for people with physical disabilities. It is to make fashion more inclusive and the design incorporates features like easy access zippers, magnetic closures, which supports users with limited mobility. Microsoft Xbox is one such example where its adaptive controller is designed for gamers with limited mobility. Its customizable design enables players to connect various assistive devices, making gaming accessible to people with different physical abilities.

Another great example of an inclusive design is Apple's Series. It can understand and respond in multiple languages and dialects. Also supporting bilingual speakers, it allows users to ask questions in more than one language without needing to switch any settings. This promotes inclusivity irrespective of user's linguistic abilities. YouTube also provides localized versions of its interface and content in over 100 countries, making it accessible to users throughout the world in their preferred language.

Now, as we end inclusivity and accessibility, we move on to design principles. These

include visibility, feedback, constraints, and affordance. If the buttons for the volume and channels are clearly visible and grouped together, it's easy for anyone to change the channels without any searching and hence improving the experience. As we move on, we see some examples of visibility. The first one being car controls are placed such that they can be easily found and used even by the drivers while driving so they can focus on the road and not take off their attention.

The second we see is actually a bad design. Well, in sensor technologies like auto faucets, users are not actually sure how to use them and have to always guess where to put their hands that will actually activate the faucets. Okay. So this is an interesting example. The car dashboard warning lights on a car dashboard for issues like low fuel, engine trouble are actually designed to be highly visible and grab the driver's attention as soon as they are lit up and hence ensures that the driver acts upon it as soon as possible.

The next we see is how iPhone allows quick access to some frequently used settings like Wi-Fi, Bluetooth, mobile data and brightness via its control center. It is highly accessible, which can be found by just swiping down or up. This keeps essential features highly visible at all times and thus improving the experience. The next we see both iOS, Android, as well as Windows and MacOS. They offer dark modes and high contrast modes.

These reduce eye strain for users and increase readability for individuals with specific visual impairments or sensitivity to bright screens. Even for users who don't have disabilities, it offers a much better experience at night as they don't need to look at white bright screens while the environment around them is pretty dark. The next feature we come across is how Kindle allow its users to adjust the text size, the font type, and the screen brightness, and hence making the reading experience more accessible for users with visual impairments. The next design principle that we see is feedback. This is actually one of the most important principles that the system should actually communicate what is happening after a user takes action.

It includes sound, highlighting, animation, and maybe even a combination of all these. Why feedback matters is that the users need to know whether their actions were successful or not. Feedback helps them understand the system's status and what to expect next. For example, When you hit a send button for an email, a feedback would come in form of a message saying email sent or a spinning wheel showing that the email is being processed. You would also think that one could show just simply a green tick that the message has been sent successfully.

Without feedback, users might get confused and try clicking multiple times or wonder if

something is wrong if they don't get a proper feedback whether the task has been successfully completed. The next design principle we have to include here is feedback. So the first example for this is the Windows Narrator, which is actually a screen reader that converts text into speech. It allows visually impaired users to interact with the Windows operating system. The design is focused on providing real-time feedback and allows users to navigate through text-heavy content using just the Windows shortcuts and thus improving the experience for impaired users.

The next example is again from Microsoft. The Microsoft Live, while you try to log in the sign-in screen, shows the strength of the passwords as the users are trying to log in. The screen displays various colors in a spectrum, which displays the strength of passwords as one continues typing. As you can see in the image here, as you keep increasing the strength of the password, the color becomes more and more green, and thus giving feedback to the users and the password is ready to send. The next we see is, this is something that Most of the users have experienced this while typing on a phone, that users receive subtle vibrations with each key press, confirming that the input has been registered.

This haptic feedback improves the tactile feel of the touch interface. Hence, it makes it more like typing on a physical keyboard than a touch screen. The next example for feedback is Microsoft Word. or any word processing software for that fact, that if a word is incorrectly spelled, misspelled, or there is some grammatical mistake, it is underlined. For grammatical mistakes, it is actually blue, while for misspellings, it is underlined in red.

This alerts the users that the word may be incorrect, prompting them to take corrective action as soon as possible. The next design principle we see is consistency. The design and behavior of similar elements should be same throughout the system. This is what defines consistency. Why it matters is that it helps users to predict how things will work and thus reducing the learning curve and preventing confusion.

The example is also pretty normal for most of you who have been using browsers for most of your time. In most apps, the back button is typically in the top left corner, and clicking it always takes you to the previous screen. If this changes from screen to screen or app to app, users would have to relearn each page and thus making the experience frustrating. How does consistency benefit? Well, first of all, it makes it easily recognizable. I mean, think of the Mercedes-Benz logo.

As you see a Mercedes car on the roads, You know that it is a Mercedes just by looking at its logo. The style and its appearance are always repeated. It makes it easy to learn and

use, and that enables the user to transfer previous knowledge to new contexts, like moving to a new platform. And well, as we saw in the last slide as well about the back button being placed on the top left corner, Back button is always present there, and thus making it pretty intuitive for you to go to the previous screen. And well, it does make it aesthetic as well.

In most website designs, majorly just two colors are used, a primary and a second color. And well, it ends up making it a little more aesthetic. And last but not the least, consistency also eliminates confusion. This example of consistency is that the yellow box, I mean, if you just have a look at this yellow box right here, the box is actually different from other boxes in shape, color, as well as the direction of the arrow. And hence, it is not consistent with the other boxes, which are blue rectangles, and the direction is to the right.

The next thing we have is constraints. Well, constraints actually limit the options available to users to reduce error or confusion. They may be of two different types, physical or psychological. Why constraints matter is it limits the users by what they can do, especially in the error prone situations. It makes it easier for them to complete their tasks without making any mistakes.

For example, when entering a password, some apps may actually prevent you from typing a password that is too short by graying out the submit button until the password meets a minimum length or a minimum criteria that includes some special characters, alphabets, or whatever. This constraint ensures that users don't submit the form prematurely with incomplete or incorrect information. The following signup interfaces actually limit the user to the available options, which reduces error or confusion. The fields like username, password, confirm your password, your name, and your date of birth may actually need you to satisfy a minimum criteria before one may proceed with the signup process, which may include, again, special characters, non-empty usernames, and password fields. And this one of the most important parts that nowadays accepting the terms and conditions.

Most apps may actually prevent you from proceeding further without clicking the accepting the terms and conditions option. The next and the last design principle we have to cover is affordance. Design elements should hint at how they can actually be used. In short, how they can be self-explanatory and the products talk for themselves. We should be able to visually see what that object or thing is being used for.

Why affordance matters? Well, if something looks like it can be used in a certain way, users are more likely to use it correctly without any extra instructions. For example, in

softwares, a button that looks raised and clickable, as opposed to some flat text, it signals that it can be pressed and thus it probably is a toggle button. Some examples for affordance, the scroll bars in various apps suggest that it can be dragged up or down to scroll through the content. The visual cue is the long vertical bar, which indicates that it can be interacted. The next we have is the elevator button, which provides us a clear visual cue that it can be pressed to request an elevator.

The button is often large and also contains arrows which indicate the direction of movement for the lift. And the buttons are also distinct in appearance. Usually they are round shaped with labels or icons which indicate its function. The next example for affordance is a hyperlink. A hyperlink on a website that is underlined and changes color when hovered over.

The visual change indicates that the hyperlink is clickable and will lead to another page. A similar example is a volume slider on audio or video players. The long horizontal bar with a movable marker visually suggests that it can be dragged left or right or for vertical bars up or down to adjust the volume or seek through the media. This is also true for the seek function in various video players where you can actually seek through the content by actually moving to the point that you need to focus on. The next example from affordance we have is a button that affords pushing.

If you see a button for electrical plants, you know that you need to push it down to turn it on. And while door knob, a round door knob, it actually affords pushing because as soon as you look at it you know that you need to turn this to open the door. When you go to certain places you see slots are being made for donations or anything of that sorts. A door handle actually affords pulling but It is a bad design because without actually giving you physical cues that whether you need to push or pull the door, it is confusing for the user on what to do.

Okay, well, this was the design principles part. And now if we move on to the in-class exercise, you need to name the design principle which is being used in each of the following scenarios. The first one is a mouse does a click sound when clicked right or left. The second one is how relative ordering of keys in a keyboard is seen across laptops from different manufacturers. The third is how the file option is placed at the top left of the toolbar in Microsoft Office. And well, this is true for all version of Microsoft Office as well.

The fourth one is a mixer grinder does not start unless the mixer jar is locked to the motor coupler of the machine. The fifth is using mouse as a paperweight. And the last one, our users cannot move the cursor beyond the physical space of the screen. Take a

minute, pause the video, and we will be right back after this, after you are done trying the answers. Okay, now we're back to the in-class exercise and we'll be discussing the solutions.

The first one is a mouse does a click sound when clicked right or left. It is an example of feedback at how the mouse actually gives you feedback after you have done clicking it. The second one is the relative ordering of keys is same throughout manufacturers. It is an example of consistency that how the order is actually consistent throughout different manufacturers. The third one is the file option is always placed at the top left of the toolbar.

in Microsoft Office. This can be looked in two different perspectives. Think of it like at the top left of the toolbar, it will actually be visible to you at all times. So it may come under visibility. However, the file option is at the top left across softwares in Microsoft Office, and hence it also comes under consistency. The fourth one is a mixer grinder does not start unless the mixer jar is locked to the motor coupler.

Well, it is an example of constraints because unless the jar is locked, it stops the motor from moving. The fifth one is using a mouse as a paperweight. Well, it is simply an affordance because as you see a mouse, you tend to keep it over a paper that you need not move. And the last one is how users cannot move the cursor beyond the physical space. Again, the users are constrained on where they can take the mouse, which can be outside the physical screen, which is typically unusable.

Well, this was the in-charge exercise, and now this is the exercise for you to complete. You need to take the application Airbnb either on the smartphone or the web version and answer the following questions. First is what is the target user base for Airbnb? And you need to evaluate Airbnb website on the design principles that you have studied in this tutorial, visibility, feedback, consistency, constraints, and affordance. And on the basis of this evaluation, you need to suggest if it is a good design or a bad design. Furthermore, what can the developers and designers do to improve Airbnb and better cater to its users? If not, how did the designers ensure that it would be catering to the user base? And you can either list them out or sketch them on a piece of paper.

Thank you so much. This was a tutorial for inclusiveness, accessibility, and the design principle.