College of Science • Intelligent Medical Systems

Human Computer Interaction

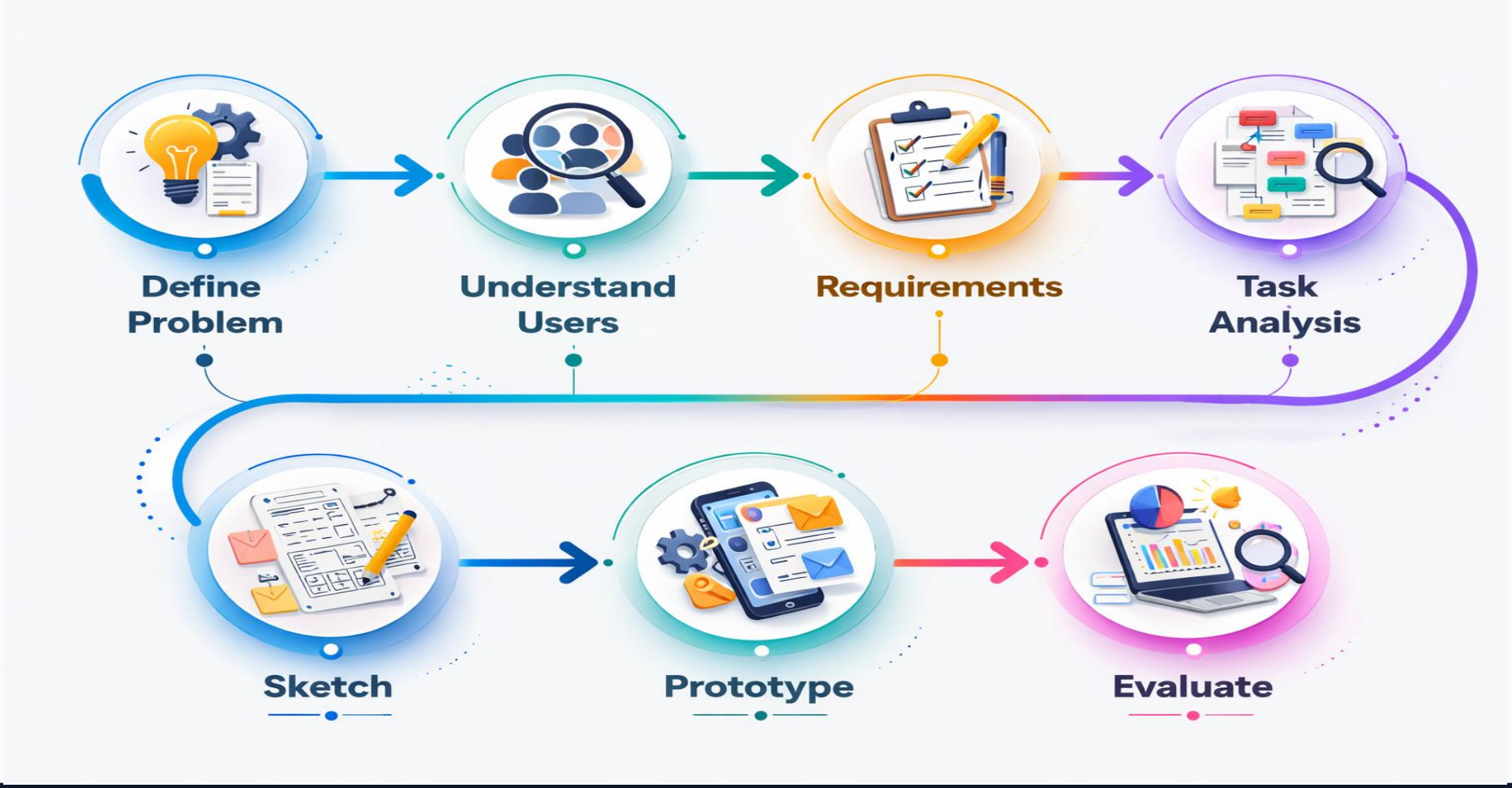
Lecture 6: Design Process for Solving Interface Problems

Asst. Lect. Ali Al-khawaja

Google Class Room

The complete HCI design process

Good interface design is not a jump from idea to code. It is a chain of decisions, and each stage reduces uncertainty before the next one begins.



Stage 1–2: Define the problem and understand users

From vague idea to real problem

Weak statement:
“We need a hospital app.”

Stronger statement:
“Reception staff need a faster and less error-prone way to register new patients during peak clinic hours.”

A useful problem statement names the user, the task, the pain point, and the context.

Questions to ask about users

- Who are the primary users?
- Are they beginners or experts?
- What are their goals?
- What mistakes do they make now?
- What pressure, noise, or time constraints affect them?

Designing for yourself instead of the target user is one of the most common student mistakes.

Stage 3 – 4: Requirements and task analysis

Functional requirements

- Register a patient
- Search for an existing record
- Update appointment details
- Print or export a summary

Usability requirements

- Easy to learn
- Low error rate
- Clear feedback
- Readable in a busy environment

Task analysis example — New patient registration

1. Open registration screen
2. Choose “new patient”
3. Enter required fields
4. Check missing or invalid data
5. Save record
6. Show success confirmation

Task analysis exposes repetition, hidden decisions, and confusing order before any visual design begins.

Stage 5–7: Goals, alternatives, and sketching

Good designers do not commit to the first interface idea. They compare alternatives and delay visual polish until the structure is right.

Form-based screen

Best for dense data entry

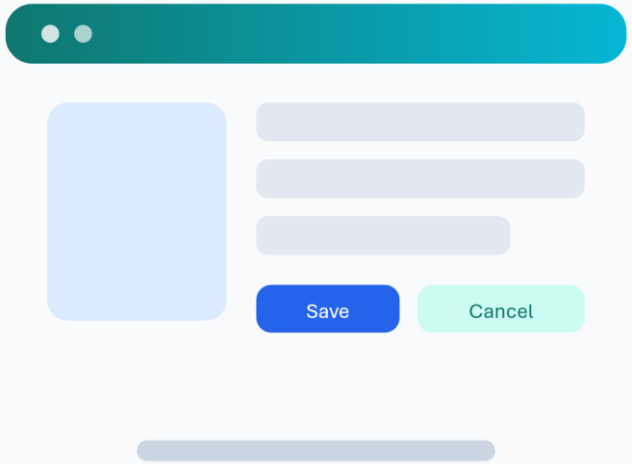
Wizard / step-by-step

Best for guided completion

Dashboard + quick actions

Best for frequent repeat tasks

Stage 8–10: Prototype, evaluate, iterate



Prototype

Build a fast preliminary version to communicate flow, layout, and interaction. Start low fidelity; polish later.

Evaluate

Check the design against visibility, consistency, error prevention, recognition, and user control.

Iterate with evidence

If users hesitate, misclick, or ask what to do next, the design is not finished. Revise layout, labels, sequence, feedback, and field grouping—then test again. In HCI, iteration is not a weakness; it is the method.

A polished but untested interface can still be unusable. Testing and redesign are what turn a screen into a usable system.

Case study: Clinic appointment booking

Problem

Patients wait too long at the desk and booking errors occur during busy periods.

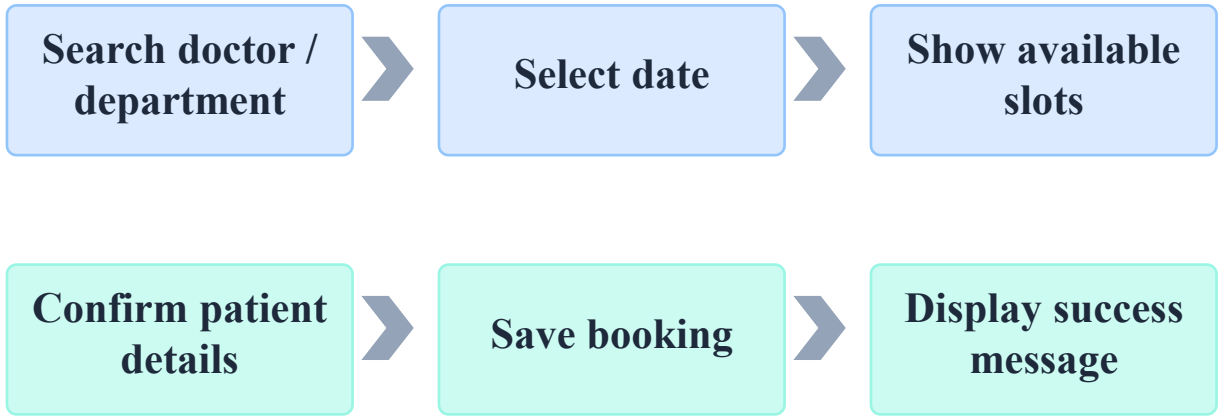
Users

Receptionist
Patient

Goals

Fast booking
Readable schedule
Clear confirmation
Prevent double booking

Task flow converted into interface logic



Notice the logic: the interface should reveal choices in the same order the user thinks about the task.

Common mistakes in student interface projects

- Starting with code before analysis
- Building only one design option
- Writing a vague problem statement
- Focusing on colors before structure
- Ignoring real users and context
- Not testing with realistic tasks
- Skipping task analysis
- Treating the first prototype as final

The hidden pattern behind all eight mistakes: students design what they can imagine quickly, not what the user can use effectively.

In-class activity and homework

Classroom task

Design a hospital reception screen for registering a new patient.

Your group must submit:

1. Problem statement
2. Target users
3. Four functional requirements
4. Four usability requirements
5. One task flow
6. One paper sketch
7. One expected usability problem

Homework

Choose one of these problems:

- Patient registration
- Appointment booking
- Student course registration
- Lab test request form

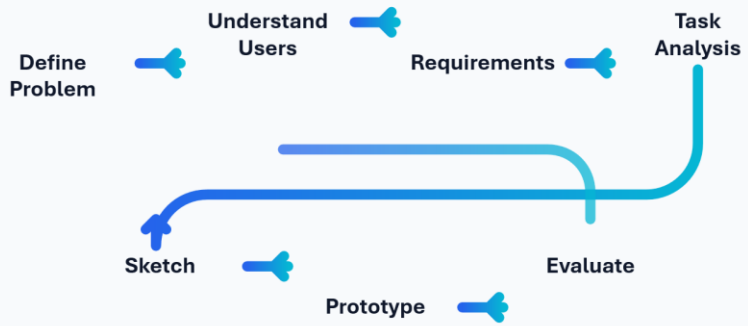
Then produce:

- Problem statement
- User analysis
- Requirements
- Task flow
- Sketch / wireframe
- Two design improvements

Summary

The HCI design process is a disciplined path from ambiguity to usability.

- Define the problem clearly before you draw screens.
- Study the users and the context they work in.
- Capture both functional and usability requirements.
- Use task analysis to organize the interface around real work.
- Generate alternatives, sketch early, and prototype cheaply.
- Evaluate with usability rules and iterate with evidence.



A good interface is rarely designed once. It is refined until the user can succeed with clarity and confidence.