# MODULE 5

## Discovering Requirements (HCI)

# Interface Types in HCI

- **Interface Types in HCI (for Discovering Requirements)**
- When we gather requirements, we also need to decide **what interface type best supports the users' tasks, goals, and environment**. Different interface types influence usability and user experience.

### 1. Command Line Interfaces (CLI)

- **Description:** Users type commands in text form.
- **Strengths:** Flexible, powerful for expert users, automation possible.
- **Weaknesses:** Steep learning curve, requires memorization.
- **Requirement Example:** A data scientist may require a CLI to run scripts efficiently.

## 2. Menu-Based Interfaces

- **Description:** Users select from a set of options (drop-down menus, pop-up menus, etc.).
- **Strengths:** Easy for beginners, reduces memory load.
- **Weaknesses:** Can be slow if too many levels; not flexible for advanced tasks.
- **Requirement Example:** ATM machines use menu-driven interfaces for simple and safe navigation.

### 3. Form-Fill Interfaces

- **Description:** Users input data into structured fields (e.g., registration forms, online booking).
- **Strengths:** Efficient for structured tasks; guides users.
- **Weaknesses:** Limited flexibility, can be tedious.
- **Requirement Example:** Hospital patient entry system requires form-filling for accuracy.

## 4. Natural Language Interfaces

- **Description:** Users interact using everyday language (typed or spoken).
- **Strengths:** Intuitive, accessible to non-technical users.
- **Weaknesses:** Ambiguity in interpretation, high processing requirements.
- **Requirement Example:** A voice assistant for elderly users in a smart home.

## 5. Direct Manipulation Interfaces

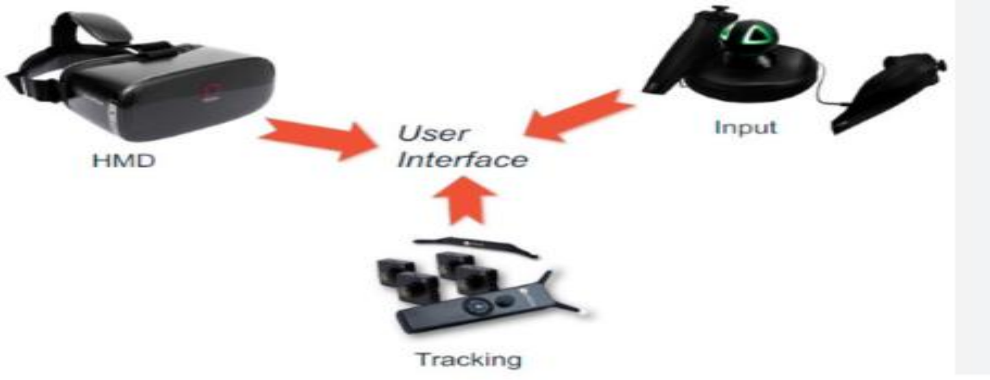
A **Direct Manipulation Interface** lets users **interact directly with digital objects** that behave like their real-world counterparts.

You can **grab, move, rotate, or modify** on-screen items **as if they were physical objects**.

**Interaction where users manipulate objects on a screen directly using input devices (mouse, touch, stylus).**

## 6. Virtual Reality (VR) Interfaces

- **Description:** Immersive 3D environments where users interact naturally.
- **Strengths:** High engagement, supports spatial tasks.
- **Weaknesses:** Expensive, motion sickness, accessibility issues.
- **Requirement Example:** VR training simulator for medical students.



### 7. Augmented Reality (AR) Interfaces

- **Description:** Overlaying digital information onto the physical world.
- **Strengths:** Context-aware, enhances real-world tasks.
- **Weaknesses:** Technical limitations, possible distraction.
- **Requirement Example:** AR navigation aid for tourists in a new city.

## 8. Tangible Interfaces

- **Tangible Interfaces** (or **Tangible User Interfaces – TUIs**) are systems that allow users to **interact with digital information through physical objects**.

- Interaction where users manipulate *physical* objects that represent digital information.
- They **bridge the gap between the physical and digital worlds** by giving digital data a **physical form** that can be touched, moved, and manipulated.

# Cont…

**Examples:**

1. Touchscreens
2. Game Controllers / VR Controllers
3. Smart Home Interfaces:

Turning a smart knob to control light brightness. Using smart buttons (like Flic) to trigger digital actions (e.g., play music).

4. Car Infotainment Knob

## 9. Conversational Interfaces (Chatbots, Voice Assistants)

- **Description:** User interacts via conversation (text or voice).
- **Strengths:** Simple, natural, available 24/7.
- **Weaknesses:** Limited understanding, may frustrate users.
- **Requirement Example:** Customer support chatbot for banking.

- When **discovering requirements**, analysing **which interface type(s)** fit the **users, tasks, and context** is essential.
- For **experts** → CLI or direct manipulation may work.
- For **novices** → Menus, forms, GUIs, or conversational interfaces are better.
- For **immersive/physical tasks** → VR, AR, or tangible interfaces fit best.

# Types of Data Gathering in HCI

- Combine multiple techniques to capture **full context**.
- Focus on **tasks, environment, user goals, and pain points**.
- Prioritize requirements based on **criticality and feasibility**.

## Data Gathering for Requirements (HCI)

- To understand **what users need** and **what the system should support**.
- Helps avoid designing based on assumptions.
- Provides both **functional** (what the system should do) and **non-functional** (usability, accessibility, performance) requirements.

## Sources of Requirements:

- **Users (End-Users, Target Audience)** – provide task-related needs.
- **Stakeholders (Managers, Clients, Domain Experts)** – define business/organizational goals.
- **Existing Systems / Documentation** – provide baseline for improvements.
- **Environment / Context of Use** – reveals practical constraints.

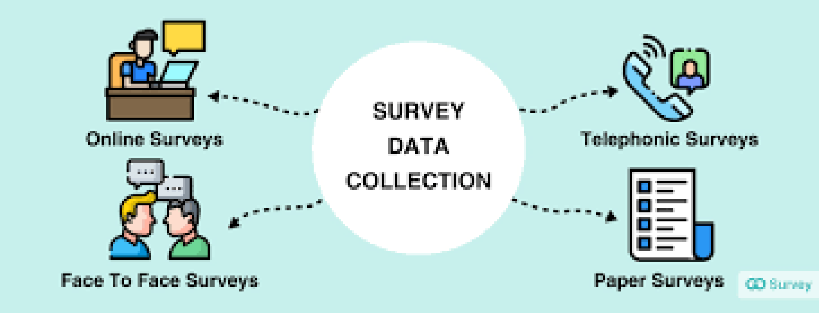
Data gathering can be broadly classified into **three categories**:

- **Interviews & Discussions**
- **Questionnaires & Surveys**
- **Observation-Based Methods**
- **Focus groups**
- **Workshops**
- **Document study, and**
- **Prototyping feedback).**

## 1. Interviews

- **Description:** One-to-one (or small group) discussions between interviewer and participant.
- **Types:**
  - *Structured* → pre-set questions, fixed order.
  - *Semi-structured* → mix of fixed and open questions.
  - *Unstructured* → free-flowing conversation.
- **Advantages:** Rich, detailed insights, clarify doubts in real-time.
- **Disadvantages:** Time-consuming, interviewer bias possible.
- **Example:** Interviewing nurses to understand problems with a patient-monitoring system.

## 2. Questionnaires / Surveys



- **Description:** Written set of questions distributed to many participants.
- **Types:**
  - *Closed-ended* (multiple choice, rating scales).
  - *Open-ended* (free text responses).
- **Advantages:** Cost-effective, covers large population, easy statistics.
- **Disadvantages:** Limited depth, low response rate possible.
- **Example:** Gathering feedback from 500 students about a university learning portal.

## 3. Observation

- **Description:** Watching users in their natural environment while they perform tasks.
- **Types:**
  - *Direct Observation* → researcher watches without interference.
  - *Participant Observation* → researcher actively joins tasks.
  - *Ethnography* → long-term observation of users in context.
- **Advantages:** Reveals real behaviors (not just what users *say*).
- **Disadvantages:** Time-consuming, may feel intrusive, observer bias.
- **Example:** Observing how clerks use an online ticketing system at a railway counter.

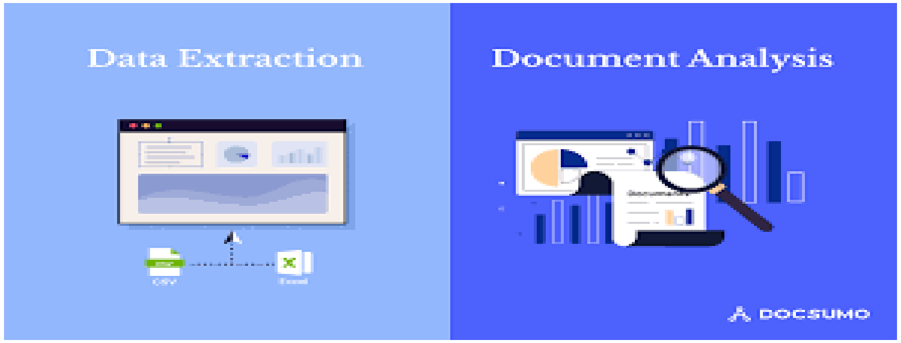
## 4. Focus Groups

- **Description:** Group discussion (6–10 people) guided by a facilitator.
- **Advantages:** Encourages brainstorming, multiple perspectives, quick feedback.
- **Disadvantages:** Dominant participants may bias discussion, less depth per individual.
- **Example:** A focus group of teachers discussing digital classroom tools.

## 5. Workshops / Brainstorming Sessions

- **Description:** Interactive sessions with multiple stakeholders for idea generation.
- **Advantages:** Fast consensus-building, collaborative.
- **Disadvantages:** Requires skilled facilitation; may lack depth.

## 6. Document Studies



- **Description:** Analyzing existing documentation, logs, manuals, or reports.
- **Advantages:** Provides background knowledge, saves time.
- **Disadvantages:** May be outdated or incomplete.
- **Example:** Studying helpdesk logs to find common software complaints.

## 7. Prototyping Feedback

- **Description:** Users interact with mock-ups or prototypes, giving direct feedback.
- **Advantages:** Concrete, helps users articulate needs.
- **Disadvantages:** Users may focus only on visual aspects, ignoring deeper issues.

| Type | Description | Advantages | Disadvantages | Example |
|---|---|---|---|---|
| **Interviews (Structured, Semi-structured, Unstructured)** | Direct conversation between researcher and participant to gather insights. | - Rich, detailed data- Clarifications possible- Flexible (esp. semi/unstructured) | - Time-consuming- Interviewer bias- Hard to generalize from small sample | Interviewing doctors to identify problems with hospital information systems |
| **Questionnaires / Surveys** | Written questions distributed to a large group (paper/online). | - Covers large population- Cost/time efficient- Easy statistical analysis | - Limited depth- Risk of low response rate- Misinterpretation possible | Surveying 500 students about usability of an e-learning portal |
| **Observation (Direct, Participant, Ethnography)** | Watching users in real context as they perform tasks. | - Reveals actual behavior (not just what users say)- Context-rich insights- Can identify hidden problems | - Intrusive if not handled carefully- Observer bias- Time-consuming | Observing railway clerks using an online ticket booking system |
| **Focus Groups** | Guided group discussion (6–10 participants) moderated by a facilitator. | - Encourages idea generation- Quick multiple perspectives- Interactive & engaging | - Dominant voices may bias results- Less individual depth- Needs skilled facilitator | Teachers in a group discussing digital classroom needs |

| Type | Description | Advantages | Disadvantages | Example |
|---|---|---|---|---|
| **Workshops / Brainstorming** | Collaborative sessions with stakeholders for requirement exploration. | - Builds consensus- Generates ideas quickly- Encourages stakeholder participation | - Can get off-track- Some participants may dominate- Needs careful facilitation | A workshop with bank employees to design a new mobile banking app |
| **Document Studies** | Analyzing existing documentation, reports, logs, manuals. | - Provides background info- Saves time & cost- Objective source of issues | - May be outdated- Not user-centered- Limited insight into real behavior | Studying helpdesk complaint logs of a telecom company |
| **Prototyping Feedback** | Users interact with mock-ups or prototypes and provide feedback. | - Concrete, easy for users to understand- Helps refine requirements early- Encourages active participation | - Users may focus only on surface aspects- May overlook deeper needs- Prototype quality affects feedback | Testing a clickable prototype of an online shopping website |

# Key Issues -
# Data Gathering in HCI

- Collecting requirements requires observing, asking, and analyzing user behavior.

**Key Issues**

- Users may **not know what they need**.
- Requirements may conflict between stakeholders.
- Environmental constraints can limit usability.
- Ensuring **accuracy and completeness** is challenging.

The **key issues in data gathering** are:

- Identifying all stakeholders.
- Choosing suitable techniques.
- Handling conflicting perspectives.
- Considering real-world context.
- Avoiding bias.
- Ensuring ethics.
- Accurate recording and analysis.
- Effective communication.

## 1. Identifying the Right Stakeholders

- Systems have multiple stakeholders: end users, managers, customers, technical staff, support teams.
- **Issue:** Missing out on important stakeholders may lead to incomplete or biased requirements.
- **Example:** Designing a hospital system without consulting nurses could ignore crucial workflow needs.

## 2. Choosing Appropriate Data-Gathering Techniques

- Each method has strengths and weaknesses.
  - Interviews → in-depth, but time-consuming.
  - Surveys → reach many people, but less detailed.
  - Observation → realistic, but may intrude.
- **Issue:** Wrong choice can produce irrelevant or misleading data.

## 3. Dealing with Different Stakeholder Perspectives

- Different groups often have **conflicting needs**.
- **Issue:** Balancing conflicting views is difficult (e.g., doctors want detailed data, patients want privacy).
- **Solution:** Prioritize based on project goals, usability, and critical tasks.

## 4. Managing Real-World Context

- The environment (social, organizational, physical) shapes user needs.
- **Issue:** If data gathering ignores context, design may not fit reality.
- **Example:** ATM design must consider users in noisy conditions.

## 5. Avoiding Bias in Data Collection

- Interviewer bias, leading questions, or selective sampling can distort findings.
- **Issue:** Collected data may reflect the researcher’s assumptions, not users’ needs.
- **Solution:** Neutral questioning, diverse participants, triangulation (multiple methods).

## 6. Ethical Considerations

- Privacy, confidentiality, informed consent are crucial.
- **Issue:** Collecting personal/sensitive data without transparency may harm trust.
- **Example:** Recording workplace conversations without consent is unethical.

## 7. Recording, Analyzing, and Interpreting Data

- Data must be **systematically documented and analyzed**.
- **Issue:** Poor recording leads to loss of insights; misinterpretation leads to wrong requirements.
- **Solution:** Use transcripts, notes, audio/video (with permission), coding schemes, affinity diagrams.

## 8. Communication with Participants

- Stakeholders may have different technical vocabularies.
- **Issue:** Misunderstanding between technical teams and end users.
- **Solution:** Use **scenarios, prototypes, storyboards** to clarify.

# Analysis, Interpretation, and Presentation

- When we gather raw data (interviews, surveys, observations, logs), it must be:
- **Analyzed** → breaking it into meaningful parts.
- **Interpreted** → making sense of what it means for design.
- **Presented** → communicating results clearly to stakeholders.

## 1. Data Analysis

### Types

- **Qualitative Analysis** (non-numeric, text/audio/video data)
  - *Coding & Categorization* → grouping similar responses.
  - *Affinity Diagramming* → clustering sticky notes into themes.
  - *Thematic Analysis* → identifying recurring patterns (e.g., usability pain points).
  - *Content Analysis* → counting keywords/phrases.

- **Quantitative Analysis** (numeric/statistical data)
  - *Descriptive Statistics* → mean, median, mode, frequency.
  - *Inferential Statistics* → hypothesis testing, correlations, regressions.
  - *Task Performance Analysis* → completion time, error rate, efficiency.

**Tools**

- **Qualitative**: NVivo, ATLAS.ti, Dedoose.
- **Quantitative**: Excel, SPSS, R, Python (pandas, NumPy, matplotlib).

## 2. Data Interpretation

- Turning analyzed results into **design implications**.
- Focus on **user goals, pain points, and contextual needs**.
- Key Questions:
  - What patterns or themes emerge?
  - What do these results mean for system design?
  - Are there conflicting needs that must be balanced?
- Example: If 60% of elderly users struggled with small text, **interpretation** = need for scalable fonts and high-contrast UI.

**Tools:**

Tableau, Power BI

## 3. Data Presentation

## Types of Presentation

- **Textual / Narrative Summaries**
  - Reports, requirement documents, design guidelines.
  - Good for detail and documentation.
- **Tabular Presentation**
  - Tables showing frequency counts, comparison of requirements.
  - Good for structured data.
- **Graphical Presentation**
  - Charts: bar graphs, pie charts, histograms.
  - Diagrams: flowcharts, Gantt charts.
  - Good for highlighting trends.

# Cont…

- **Visual & Interactive Models**
    - *Personas* → fictional user profiles with goals/traits.
    - *Use Cases / Scenarios* → step-by-step task descriptions.
    - *Storyboards* → comic-like sequence of user interactions.
    - *Wireframes / Prototypes* → visual system representation.
    - Good for engaging stakeholders.
- 🔹 **Tools**
- **Charts & Statistics**: MS Excel, Google Sheets, SPSS, Tableau, Power BI.
- **Visual Models**: UML tools, Figma, Balsamiq, Miro, Lucidchart.
- **Reports**: MS Word, LaTeX.

# Ethical Design Concerns

- Protect **user privacy** and sensitive data.
- Avoid **bias** in data collection and interface design.
- Ensure **accessibility and inclusiveness**.
- Maintain **transparency** about data usage.
- **Example:** In medical apps, anonymize patient data and get consent for usage.

## Ethical Design Concerns in HCI:

- Ethical issues arise mainly in **data gathering, analysis, and system design**. They ensure respect for **users' rights, dignity, and fairness**.

**1. Informed Consent**

- Participants must know **what data is being collected, why, how it will be used, and who will access it**.
- Consent must be **voluntary**, not forced.
- **Example:** Before recording a usability test, clearly inform participants and get signed consent.

## 2. Privacy & Confidentiality

- Protect sensitive personal information (name, health records, location).
- Anonymize data where possible.
- Store securely to prevent leaks.
- **Example:** In a medical app study, remove patient identifiers before analysis.

## 3. Avoiding Harm

- Research/design should not cause **physical, emotional, or psychological harm**.
- Avoid tasks that cause stress, embarrassment, or fatigue.
- **Example:** Do not ask elderly participants to perform overly complex stressful tasks.

## 4. Bias and Fairness

- Ensure inclusivity: avoid bias based on **age, gender, culture, ability, or socio-economic status**.
- Systems must not unfairly disadvantage groups.
- **Example:** A recruitment system should not favor one gender due to biased training data.

## 5. Accessibility and Inclusiveness

- Designs must consider **disabled users, elderly, and diverse groups**.
- Ethical to design for *universal usability*.
- **Example:** Providing screen reader support, high contrast modes, captions for videos.

## 6. Transparency

- Users should understand how the system works (avoid “black box” design).
- Be open about data usage, algorithms, and limitations.
- **Example:** An AI loan approval system should explain *why* an application was rejected.

## 7. Ownership of Data

- Respect intellectual property and ownership of user-generated data.
- **Example:** A photo-sharing app should not sell user photos without permission.

## 8. Voluntary Participation & Withdrawal

- Participants should be free to withdraw at any stage without penalty.
- **Example:** A user in a study should be able to stop answering a survey anytime.

## 9. Cultural Sensitivity

- Designs should respect cultural values, norms, and practices.
- **Example:** Color choices (e.g., white for mourning in some cultures) should not offend users.

# Capturing Interaction with Use Cases

- In Human–Computer Interaction (HCI), one of the key challenges is understanding how users will interact with a system.
- Use Cases are a structured way of describing these interactions in terms of goals, tasks, and system responses.
- They capture requirements from the user’s perspective, making them more relatable and easier to validate than technical specifications.

## What is a Use Case?

- A use case is a scenario-based description of how an actor (a user or external system) interacts with the system to achieve a goal.
- It focuses on user tasks and system support, not on internal details of implementation.
- It helps answer:
  - *What do users want to achieve?*
  - *How will they interact with the system to achieve it?*

**Components of a Use Case:**

A standard use case template contains:

- Use Case ID/Name – Unique and descriptive.
- Actors – Who interacts with the system? (users, administrators, external systems).
- Goal/Purpose – What the actor wants to achieve.
- Preconditions – Conditions that must be true before the use case starts.
- Trigger – The event/action that initiates the use case.

# Cont…

- Main Success Scenario (Basic Flow) – Step-by-step normal interaction.
- Alternative/Exception Flows – Variations if things go wrong (errors, invalid data, system failure).
- Postconditions – What must be true when the use case ends successfully.
- Priority/Notes (optional) – Importance of the use case, constraints, usability concerns.

## Example in Detail (Online Library System)

- **Use Case Name:** Borrow a Book
- **Actors:** Student (primary), Library Database (secondary system)
- **Goal:** To borrow a book online.
- **Preconditions:**

  Student must be registered and logged in.

  The library database must be available.
- **Trigger:** Student clicks the "Borrow" button.

- **Main Flow:**

  Student searches for a book.

  System displays a list of available books.

  Student selects a book and clicks "Borrow."

  System checks the book’s availability.

  System records the transaction and updates the database.

  System displays a confirmation message.

- **Alternative Flow (Unsuccessful Borrow):**

  1. If the book is unavailable → System shows "Not available, join waitlist."

  2. If student account has exceeded borrowing limit → System displays "Borrowing limit reached."

- **Postconditions:**

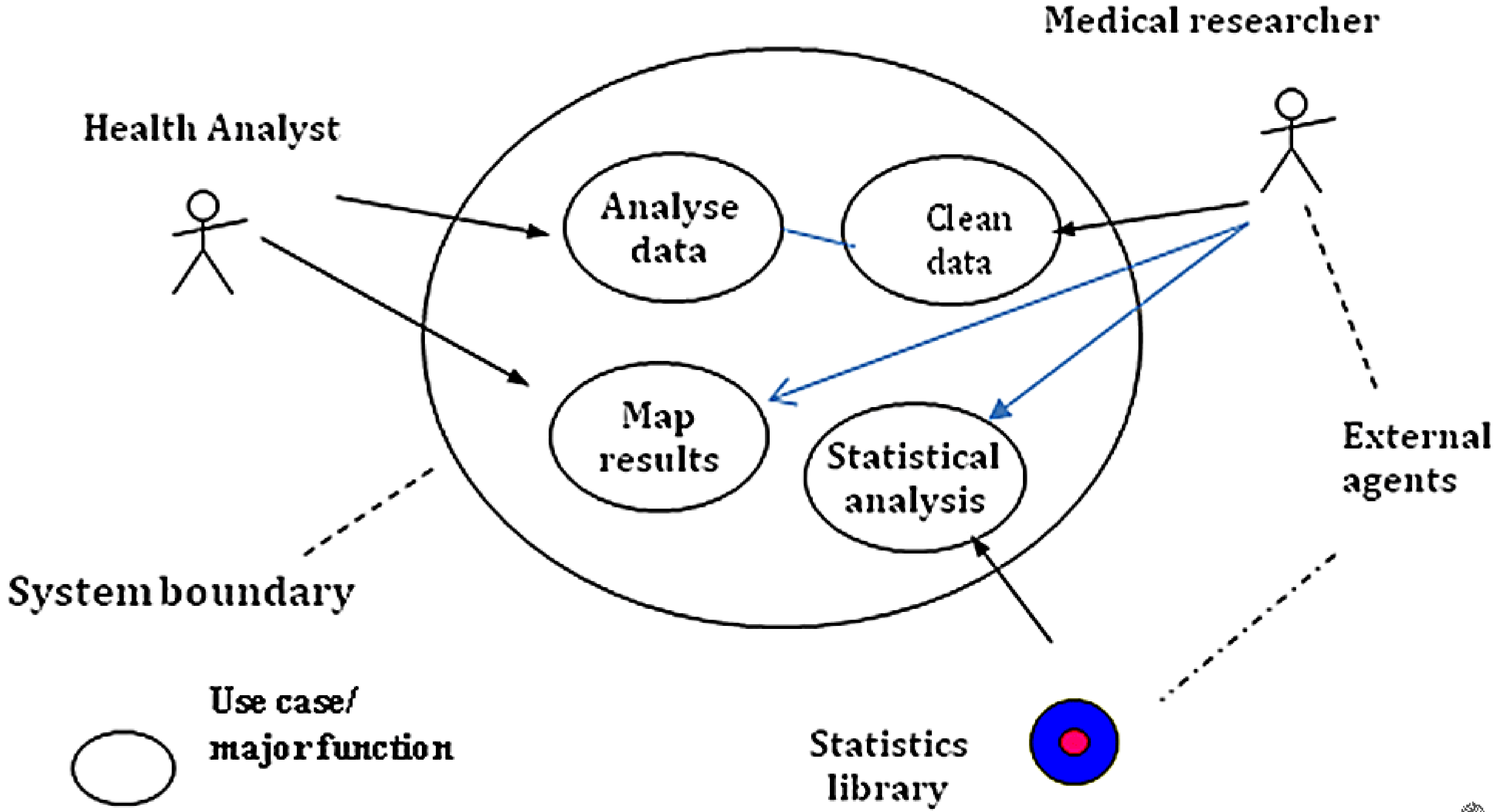
  1. If successful, book is issued and linked to student’s account.

  2. If unsuccessful, no changes to borrowing records.

## Benefits of Use Cases in HCI

- User-Centered – Keeps focus on user tasks, not just system functions.
- Requirement Discovery – Makes hidden requirements visible.
- Communication Tool – Easy for both developers and non-technical stakeholders to understand.
- Foundation for Design – Supports storyboards, prototypes, and interface flows.
- Basis for Testing – Test cases can be derived from use cases.

# Ex:


Medical researcher
Health Analyst
Analyse data
Clean data
Map results
Statistical analysis
External agents
System boundary
Use case/ major function
Statistics library