Step 1 - Understand the problem and establish design scope

> Goal: collect requirements(what we have and what we want)
>
> - Ask the right questions
> - make the proper assumptions(write down)

Don't be like Jimmy!

Example

If you are asked to design a news feed system, you want to ask questions that help you clarify the requirements. The conversation between you and the interviewer might look like this:

Candidate: Is this a mobile app? Or a web app? Or both?

Interviewer: Both.

Candidate: What are the most important features for the product?
Interviewer: Ability to make a post and see friends' news feed.

Candidate: Is the news feed sorted in reverse chronological order or a particular order? The particular order means each post is given a different weight. For instance, posts from your close friends are more important than posts from a group.
Interviewer: To keep things simple, let us assume the feed is sorted by reverse chronological order.

Candidate: How many friends can a user have?
Interviewer: 5000

Candidate: What is the traffic volume?
Interviewer: 10 million daily active users (DAU)

Candidate: Can feed contain images, videos, or just text?
Interviewer: It can contain media files, including both images and videos.

Strategies for asking questions

- General → specific
- Use-case driven
- Workflow-based

- Role-based(end-users, administrators, and other system integrators)

Questions could be asked to clarify problems

1. **Purpose and Core Features:**
    - What is the primary purpose of the system/application?
    - What are the core features that the system must support?
2. **User Base:**
    - Who are the intended users of the system?
    - How many users do we expect to serve, both initially and as we scale?
3. **Usage Patterns:**
    - What will be the typical usage patterns? Are there expected peak times?
    - How frequent are the interactions expected to be from the users?
4. **Data:**
    - What types of data will the system handle?
    - What is the expected size and growth rate of the data?
    - Are there specific data retention policies or privacy concerns to consider?
5. **Performance Requirements:**
    - What are the latency and throughput requirements?
    - Are there any real-time processing or response requirements?
6. **Scalability:**
    - Does the system need to scale horizontally or vertically?
    - What are the scalability expectations in terms of user load or data volume increase?

    ==========

    (Advanced, not limited to interviews)
7. **Integration:**
    - Are there any external systems or services with which this system needs to integrate?
    - What are the dependencies on external APIs or third-party services?

8. **Availability and Reliability:**
    - What is the desired uptime for the system? Are there different requirements for different parts of the system?
    - How fault-tolerant does the system need to be?
9. **Security Requirements:**
    - What level of security is required?
    - Are there specific compliance or regulatory requirements that need to be met?
10. **Budget and Resources:**
    - What are the constraints on budget and resources?
    - Is there a preference for certain technologies based on existing expertise or licenses?
11. **Maintenance and Monitoring:**
    - What are the expectations around system maintenance and updates?
    - How should the system be monitored, and what metrics are important to track?
12. **Timeline:**
    - What are the key milestones for the project?
    - Is there a specific timeline or deadline for having the initial version of the system operational?

Example: Online Bookstore

Context:
You are tasked with designing an online bookstore that allows users to browse books, add them to a cart, and purchase them. The system should handle user accounts, book inventory management, and process transactions.

Requirements:

1. **User Interface**:
    - Users should be able to search for books by title, author, or genre.
    - Users should be able to view book details including price, author, summary, and reviews.
    - Users should be able to create and manage their accounts.

2. **Core Features**:
    - Shopping cart functionality.
    - A checkout process that includes payment processing.
    - Review and rating system for books.
3. **Administration**:
    - The ability for administrators to add, remove, or update book listings.
    - Reporting tools for sales, inventory levels, and user activity.
4. **Scalability and Performance**:
    - The system should support thousands of concurrent users.
    - The system should ensure quick response times even under heavy load.
5. **Security and Compliance**:
    - Secure handling of user data and transactional information.
    - Compliance with data protection regulations.

Objective:
Design a scalable, reliable, and efficient system to meet the needs of both customers and administrators, considering future growth and potential high-demand periods like holiday seasons.

Questions could be asked:

1. **Purpose and Core Features:**
    - What specific features are most critical for the online bookstore (e.g., search functionality, recommendations)?
2. **User Base:**
    - Who is our target audience (e.g., general public, specific interest groups)?
    - Do we expect user traffic to vary seasonally or with specific promotions?
3. **Usage Patterns:**
    - What are the peak traffic expectations, and how often do they occur?
    - What is the expected frequency and volume of transactions per user?
4. **Data:**

- What kinds of data will we be handling (e.g., user data, payment information, book details)?
- How much data storage will be required initially, and what is the expected growth over time?

5. **Performance Requirements:**
    - What are the expected response times for user queries and transactions?
    - Are there specific performance metrics we need to meet for search results and checkout processes?
6. **Scalability:**
    - How will the system scale to handle thousands of concurrent users or more?
    - What strategies will we use for scaling, such as cloud services, load balancers, or database sharding?
7. **Integration:**
    - What external services will we need to integrate, such as payment gateways or social media platforms for login?
    - Are there existing inventory systems or databases that need to be integrated?
8. **Availability and Reliability:**
    - What uptime is required for the bookstore to be considered reliable?
    - What redundancy plans and backup systems will be in place to ensure data integrity?

==========

(Advanced, not limited to interviews)

9. **Security Requirements:**
    - What levels of security are needed for user data and financial transactions?
    - How will we comply with regulations like GDPR or PCI DSS?
10. **Budget and Resources:**
    - What is the budget for developing and maintaining the online bookstore?
    - What resources do we have available, and what might we need to acquire (e.g., technology, personnel)?
11. **Maintenance and Monitoring:**

- What tools will we use to monitor the health of the system and user activity?
- How will we handle updates and maintenance without affecting availability?

12. **Timeline:**
    - When does the system need to be operational?
    - Are there key milestones, such as beta launches or marketing campaigns, that affect the timeline?