Types of Testing and Test Cases

UNIT IV

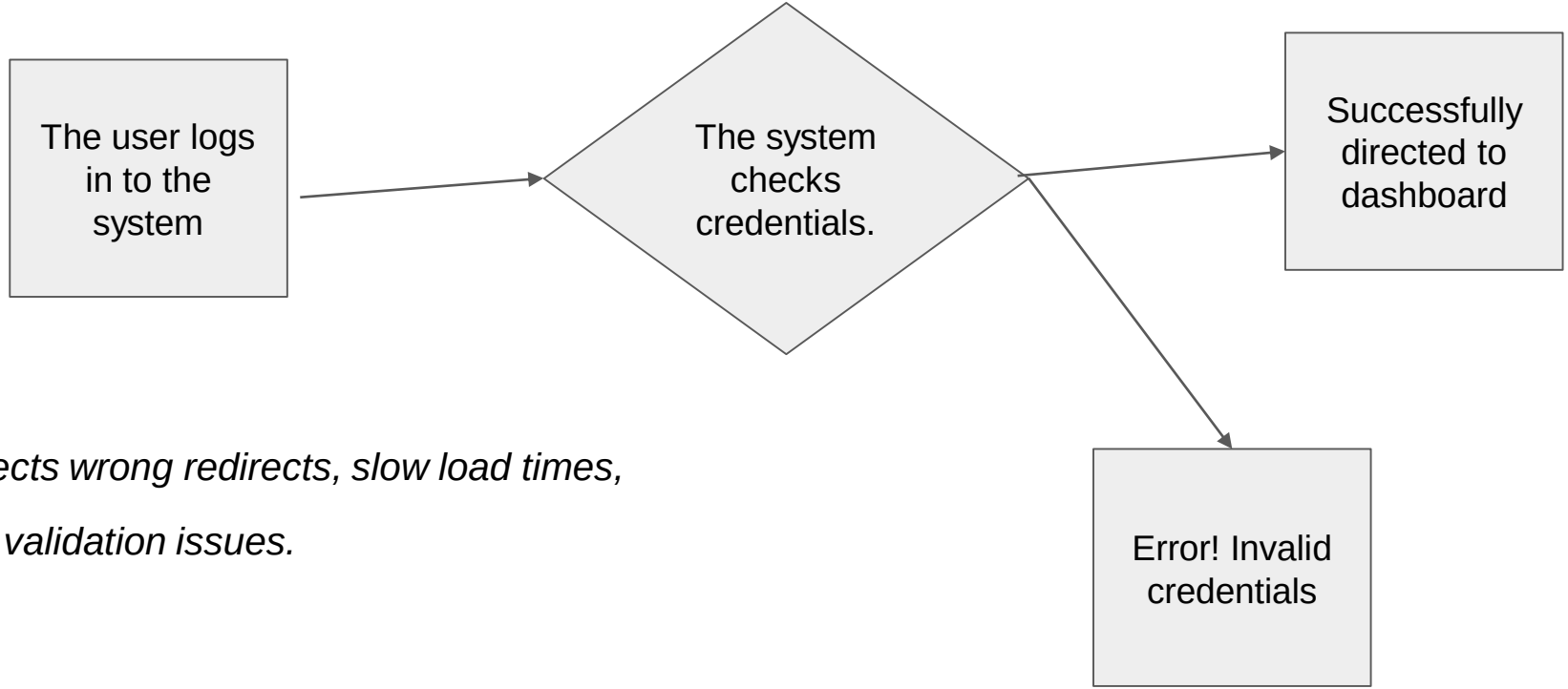
Dynamic Testing

- Testing that **requires the execution of code** to detect defects.
- Goal: Validate functionality, performance, behavior, and stability during run time.
- Where Used: Later phases of SDLC (after build).
- Key Activities:
  - Running test cases
  - Observing outputs
  - Error detection through execution

Dynamic Testing Characteristics

- Executed on working software
- Detects run-time errors (null pointer, memory leaks, incorrect results)
- Helps evaluate performance, usability, functional correctness
- Involves test environments resembling real systems
- Includes Functional, Non-functional, and Security testing

Dynamic Testing Example



Detects wrong redirects, slow load times,

and validation issues.

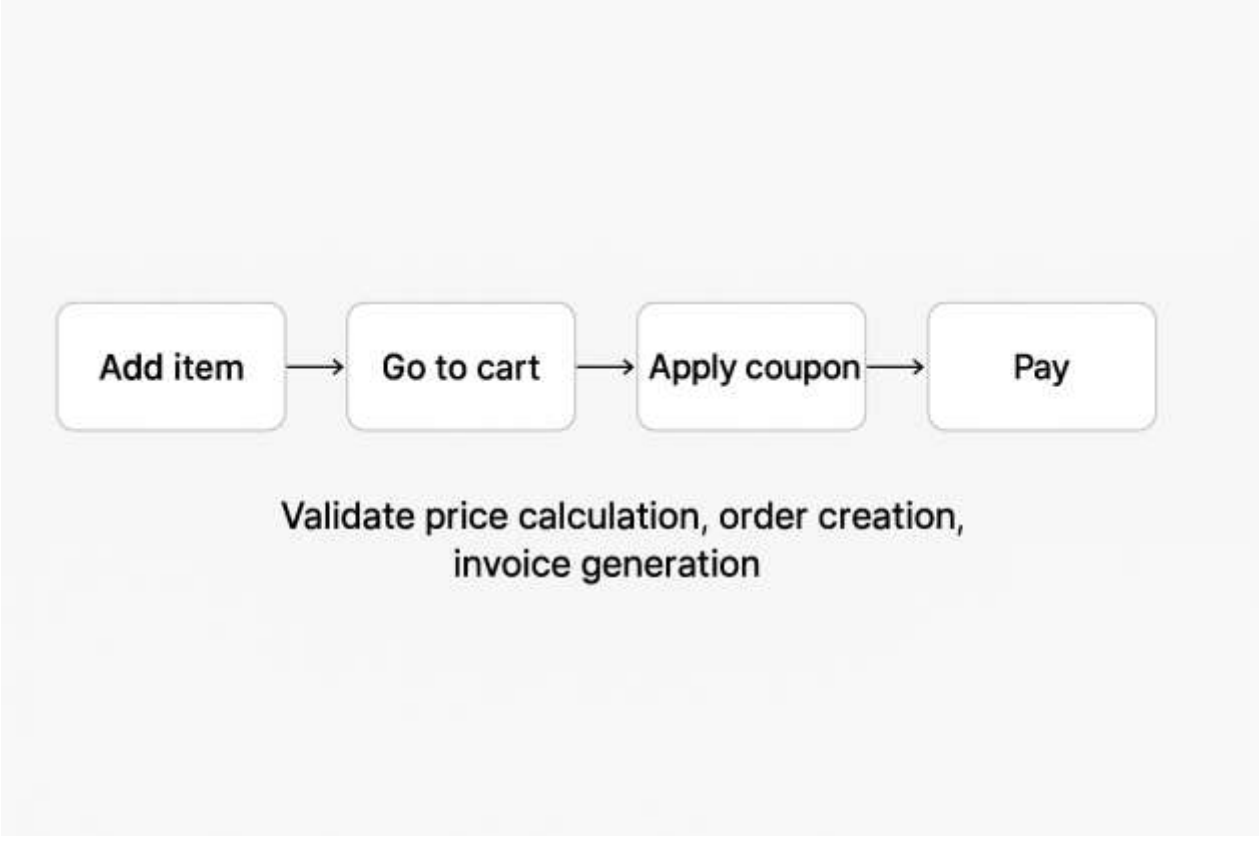
FUNCTIONAL TESTING (TESTING OF FUNCTIONS)

- Ensures the system works according to the specified requirements.
- Based on input → output behavior.
- The tester does not bother about internal code logic.
- answer the questions like “can the user do this” or “does this particular feature work”.

Functional Testing Perspectives

- Requirement-based testing
  - Prioritize requirements based on risks
  - Run critical tests first
  - Eg: user login is always the top priority
- Business-process-based testing
  - How businesses use system in daily work
  - Eg: Employee registration workflow
    - New employee joins
    - Admin creates employee account
    - Employee should be able to access office resources
  - Eg: Ecommerce workflow

Functional Testing Example



Functional Testing Types

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing (UAT)
- Smoke Testing
- Regression Testing