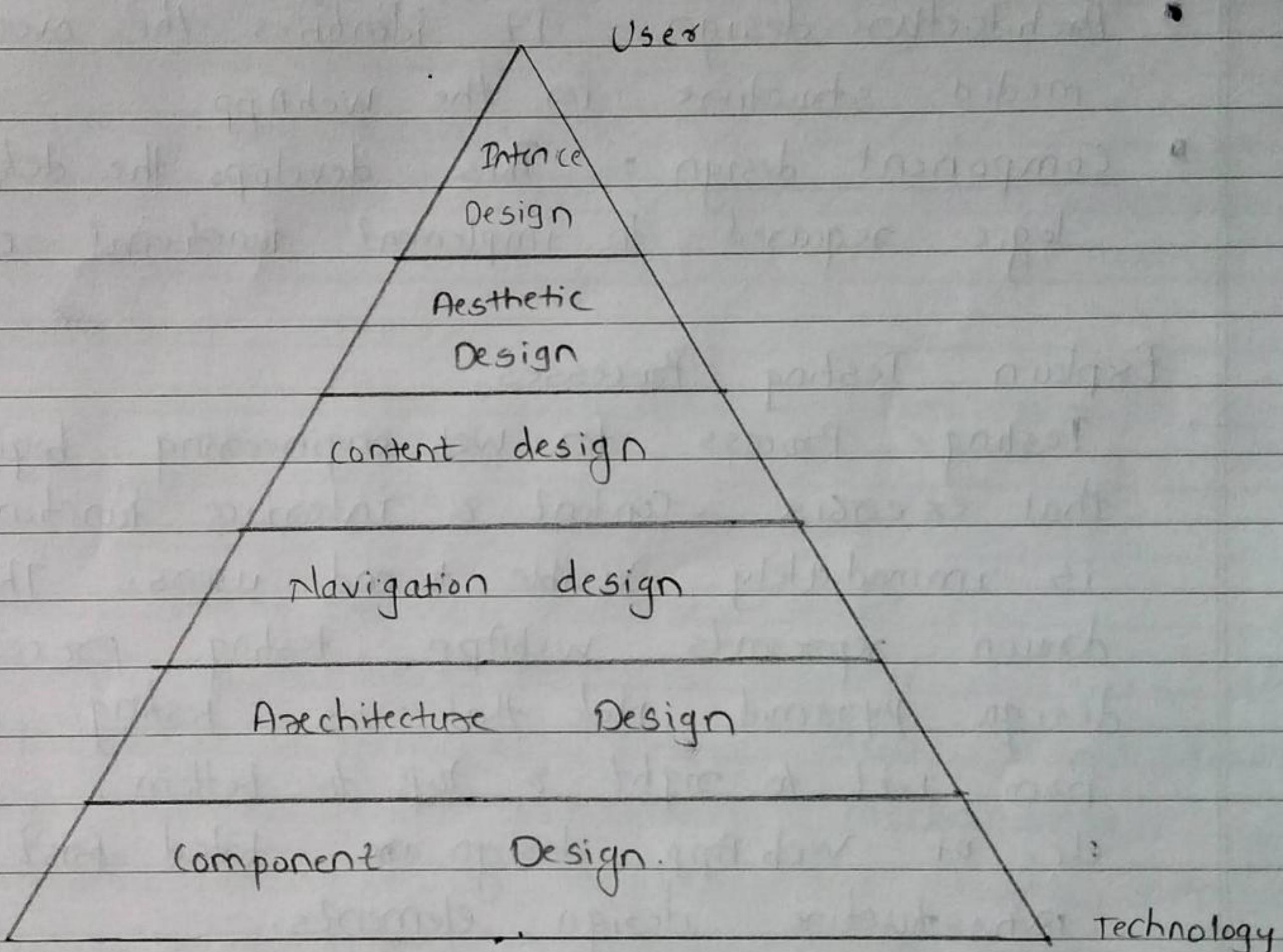


Q. 1. Explain webE design Pyramid.

→ Design leads to a model that contains the appropriate mix of aesthetics, content & technology. The mix will vary depending upon nature of webApp & as consequence design activities emphasized will also vary. Below given fig. depicts design pyramid for web engg. Each level of pyramid represents one of the foll. design activities:



- Interface Design:- It describes structure & organⁿ of the user interface. It includes representation of screen layout, a defⁿ of modes of interaction & descriptⁿ of navigatⁿ mechanisms.
- Aesthetic Design:- Also called graphic design. describes "look & feel" of the webApp of the webApp. Includes colour schemes, geometric layout,

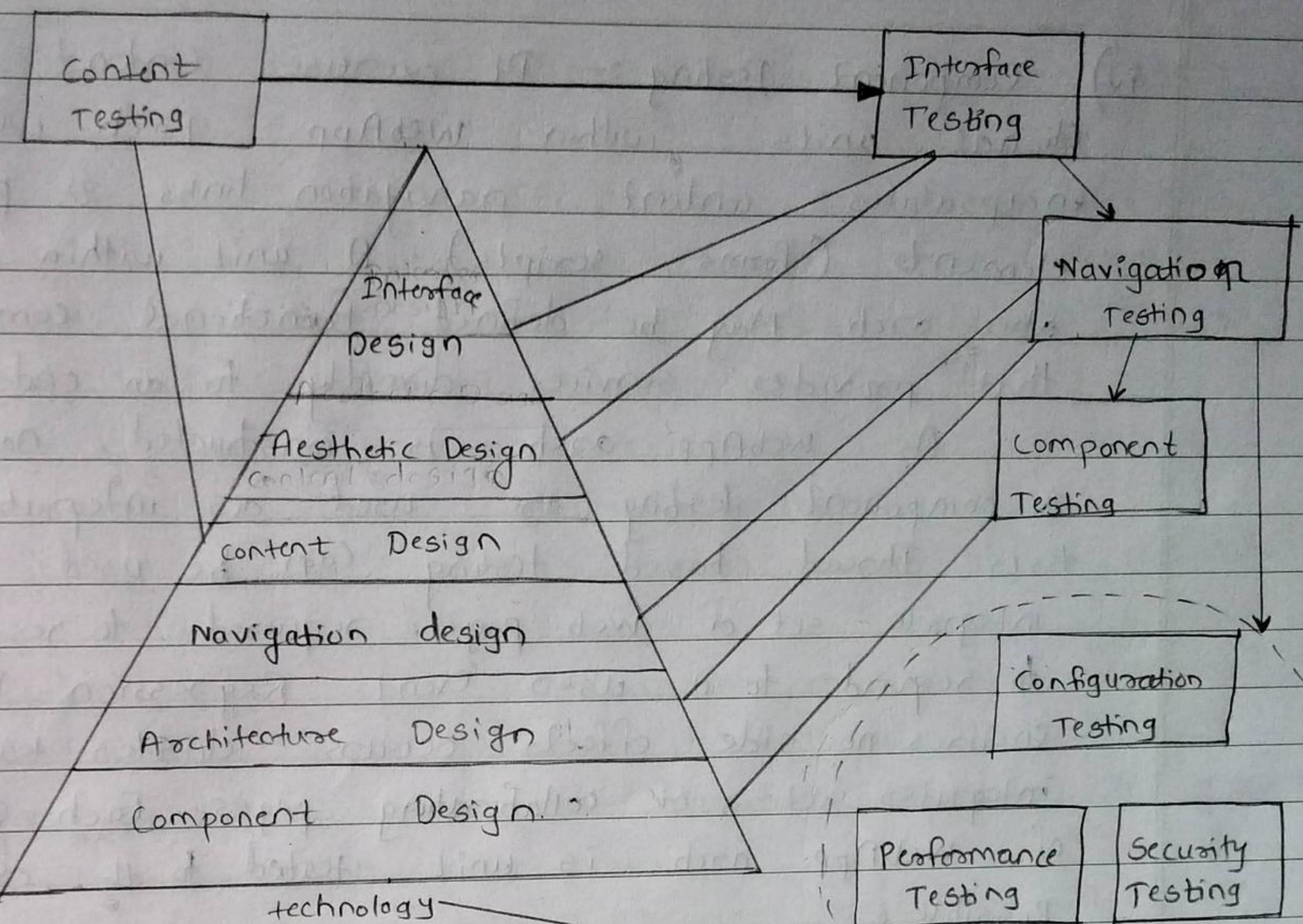
text-size, font & placement, use of graphics & related aesthetic decisions.

- Content design :- It defines layout, structure & outline for all content that is presented as part of the WebApp. Establishes the relationships betⁿ content objects.
- Navigation design : Represents navigational flow betⁿ content objects & for all webApp function.
- Architecture design :- It identifies the overall hyper-media structure for the WebApp.
- Component design :- This develops the detailed processing logic required to implement functional components.

2. Explain Testing Processes.

→ Testing Process for web engineering begins the tests that exercise content & interface functionality that is immediately visible to end-users. The fig. drawn represents webApp testing process with design pyramid. Note that as testing proceeds from left to right & top to bottom, user visible ele. of web App design are tested first, followed by infrastructure design elements.

- i) Content Testing : It attempts to uncover errors in content. It is similar to copy-editing for a written document. In fact large website might enlist the services of a professional copy editor to uncover typographical errors, grammatical errors, errors in content consistency, graphical representation & cross referencing errors.



- ii) Interface Testing :- It exercises interface mechanism & validates aesthetics aspects of user interface. The intent is to uncover errors that result from poorly implemented interaction mechanisms.
- iii) Navigation Testing :- It applies use-cases, derived as part of the analysis activity in design of test cases that exercise each usage scenario against navigation design. Navigation mechanisms implemented within interface layout are tested against use-cases to ensure any errors that impede completion of use-case identified are corrected.

iv) Component Testing :- It exercises content & functional units within WebApp. Each web page encapsulates content, navigation links & processing elements (forms, scripts). A unit within webApp arch. May be defined functional component that provides services directly to an end-user.

As WebApp arch. is constructed, navigat & component testing are used as integration tests. Thread based testing can be used to integrate set of web pages required to respond to respond to a user event. Regression testing ensures no side effects occurs. Clustered testing integrates set s. of collaborating pages. Each ele. of WebApp arch. is unit tested to the extent possible.

v) Configuration Testing :- It attempts to uncover errors that are specific to particular client or server environment.

vi) Security Testing :- It cooperates series of tests designed to exploit vulnerabilities in webApp & it's environment. Intent is to demonstrate that security breach is possible.

vii) Performance Testing :- It encompasses a series of test that are designed to assess how webApp response time & reliability are affected by increased user traffic & how performance degradation impacts overall webApp objectives & requirements.

Q. 3. Describe content testing & user interface testing in brief.

- ■ Content Testing :- It attempts to uncover minor typographical errors or incorrect info & many other problems before the user encounters them.
 - Content testing has 3 imp objectives :
 - i) to uncover errors in text-based docs, graphical representations & other media.
 - ii) to uncover semantic errors in any object presented as navigation occurs.
 - iii) To find errors in organization or structure of content that is presented to the end-user.
 - To accomplish 1st objective automated spelling & grammar checkers may be used.
 - Semantic testing focuses on info. presented within each content object. The tester must ans q's like :-
 - i) Is the info. faculty accurate?
 - ii) Is the layout of the content object easy for the user to understand?
 - iii) Is the layout of the content object easy for the user to understand?
 - iv) Is content offensive, misleading or does it open the door to litigation?
 - v) Does content contain internal links that supplement existing content?
 - vi) Does content infringe on existing copyrights?
 - vii) Does the aesthetic style of the content conflict with the aesthetic style of the interface?
- During content testing the structure & organisation

of the content architecture is tested to ensure that required content is presented to the end-user in proper order & relationships.

■ User Interface Testing :

- ① Interface Testing Strategy :- Overall strategy for interface testing is to uncover errors related to specific interface mechanisms & also to uncover errors in way interface implements semantics of navigation, WebApp functionality or content display. The interface Testing Mechanisms are :
- ② links : Navigation mechanisms that link user to some other content object or function.
- ③ Forms : Structured doc containing blank fields that are filled by users. The data contained in the fields are used as input to one or more webApp function.
- ④ Client-side Scripting : list of program commands that in a scripting lang. that handle info i/p.
- ⑤ Dynamic HTML :- Leads to content objects that are manipulated on client side using Scripting or CSS.
- ⑥ Streaming Content :- Rather than waiting for request from client-side, content obj are downloaded automatically from the server side.
- ⑦ CGI Scripts :- common Gateway Interface Scripts implements a standard method that allows web servers to interact.
- ⑧ Cookies :- a block of data sent by the server & stored by a browser as a consequence of specific user interaction.

ii) Usability Tests :- Are designed by webE team and executed by end users.

Testing Sequences →

- define set of usability testing categories & identify goals.
- design tests that will enable each goal to be evaluated
- Select participants who will conduct tests
- Develop mechanism for accessing usability of webApp.

iii) Compatibility Testing :- It defines set of "commonly encountered" client side computing configuration & their variations.

- Create a tree structure identifying each computing platform, typical display devices, as supported on platform, the browsers available, etc.

iv) Component - level Testing :- Focuses on sets of tests that attempts to uncover errors in webApp function.

- Conventional black box & white-box test case design methods can be used.