Unit- IV
Testing Specialized systems and applications

- Dr. Shivani Budhkar

Testing Object oriented software

- Software typically undergoes many levels of testing, from unit testing to system or acceptance testing. Typically, in-unit testing, small "units", or modules of the software, are tested separately with focus on testing the code of that module.
- As information systems are becoming more complex, the object-oriented paradigm is gaining popularity because of its benefits in analysis, design, and coding. Conventional testing methods cannot be applied for testing classes because of problems involved in testing classes, abstract classes, inheritance, dynamic binding, message, passing, polymorphism, concurrency, etc.
- Testing classes is a fundamentally different problem than testing functions. A function (or a procedure) has a clearly defined input-output behavior, while a class does not have an input-output behavior specification. We can test a method of a class using approaches for testing functions, but we cannot test the class using these approaches.

Testing Object oriented software

- According to Davis the dependencies occurring in conventional systems are:
  - Data dependencies between variables
  - Calling dependencies between modules
  - Functional dependencies between a module and the variable it computes
  - Definitional dependencies between a variable and its types.
- But in Object-Oriented systems there are following additional dependencies:
  - Class to class dependencies
  - Class to method dependencies
  - Class to message dependencies
  - Class to variable dependencies
  - Method to variable dependencies
  - Method to message dependencies
  - Method to method dependencies

Issues in Testing Classes

- Additional testing techniques are, therefore, required to test these dependencies.
- Another issue of interest is that it is not possible to test the class dynamically, only its instances i.e, objects can be tested.
- Similarly, the concept of inheritance opens various issues e.g., if changes are made to a parent class or superclass, in a larger system of a class it will be difficult to test subclasses individually and isolate the error to one class.
- In object-oriented programs, control flow is characterized by message passing among objects, and the control flow switches from one object to another by inter-object communication. Consequently, there is no control flow within a class like functions. This lack of sequential control flow within a class requires different approaches for testing. Furthermore, in a function, arguments passed to the function with global data determine the path of execution within the procedure. But, in an object, the state associated with the object also influences the path of execution, and methods of a class can communicate among themselves through this state because this state is persistent across invocations of methods. Hence, for testing objects, the state of an object has to play an important role.

Techniques of object-oriented testing

- **Fault Based Testing:**
  This type of checking permits for coming up with test cases supported the consumer specification or the code or both. It tries to identify possible faults (areas of design or code that may lead to errors.). For all of these faults, a test case is developed to “flush” the errors out. These tests also force each time of code to be executed. This method of testing does not find all types of errors. However, incorrect specification and interface errors can be missed. These types of errors can be uncovered by function testing in the traditional testing model. In the object-oriented model, interaction errors can be uncovered by scenario-based testing. This form of Object oriented-testing can only test against the client’s specifications, so interface errors are still missed.
- **Class Testing Based on Method Testing:**
  This approach is the simplest approach to test classes. Each method of the class performs a well defined cohesive function and can, therefore, be related to unit testing of the traditional testing techniques. Therefore all the methods of a class can be involved at least once to test the class.
- **Random Testing:**
  It is supported by developing a random test sequence that tries the minimum variety of operations typical to the behavior of the categories
- **Partition Testing:**
  This methodology categorizes the inputs and outputs of a category so as to check them severely. This minimizes the number of cases that have to be designed.
- **Scenario-based Testing:**
  It primarily involves capturing the user actions then stimulating them to similar actions throughout the test.
  These tests tend to search out interaction form of error.

Object-Oriented Metrics

- Metrics can be broadly classified into three categories:
- **Project Metrics**
- Project Metrics enable a software project manager to assess the status and performance of an ongoing project. The following metrics are appropriate for object-oriented software projects −
    - Number of scenario scripts
    - Number of key classes
    - Number of support classes
    - Number of subsystems
- **Product Metrics**
- Product metrics measure the characteristics of the software product that has been developed. The product metrics suitable for object-oriented systems are
- **Methods per Class** − It determines the complexity of a class. If all the methods of a class are assumed to be equally complex, then a class with more methods is more complex and thus more susceptible to errors.
- **Inheritance Structure** − Systems with several small inheritance lattices are more well–structured than systems with a single large inheritance lattice. As a thumb rule, an inheritance tree should not have more than 7 (± 2) number of levels and the tree should be balanced.

Object-Oriented Metrics

- **Coupling and Cohesion** – Modules having low coupling and high cohesion are considered to be better designed, as they permit greater reusability and maintainability.
- **Response for a Class** – It measures the efficiency of the methods that are called by the instances of the class.
- **Process Metrics**
- Process metrics help in measuring how a process is performing. They are collected over all projects over long periods of time. They are used as indicators for long-term software process improvements. Some process metrics are –
- Number of KLOC (Kilo Lines of Code)
- Defect removal efficiency
- Average number of failures detected during testing
- Number of latent defects per KLOC

Testing Web Applications

- **WEB TESTING**, or website testing is checking your web application or website for potential bugs before its made live and is accessible to general public. Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website.
- During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.
- A web-based system needs to be checked completely from end-to-end before it goes live for end users.
- By performing website testing, an organization can make sure that the web-based system is functioning properly and can be accepted by real-time users.

Testing types/technique for web applications/websites

- Testing types/technique may be performed depending on your web testing requirements.
  - Functionality Testing
  - Usability testing
  - Interface testing
  - Compatibility testing
  - Performance testing
  - Security testing

Functionality Testing of a Website

- Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing, etc.
- **Check all the links:**
  - Test the outgoing links from all the pages to the specific domain under test.
  - Test all internal links.
  - Test links jumping on the same pages.
  - Test links used to send email to admin or other users from web pages.
  - Test to check if there are any orphan pages.
  - Finally, link checking includes, check for broken links in all the above-mentioned links.
- **Test forms on all pages:**
  Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So what should be checked in these forms?
  - First, check all the validations on each field.
  - Check for default values of the fields.
  - Wrong inputs in the forms to the fields in the forms.
  - Options to create forms if any, form delete, view or modify the forms.

Functionality Testing of a Website

- **Cookies Testing:**
  - Cookies are small files stored on the user machine. These are basically used to maintain the session- mainly the login sessions. Test the application by enabling or disabling the cookies in your browser options.
  - Test if the cookies are encrypted before writing to the user machine. If you are testing the session cookies (i.e. cookies that expire after the session ends) check for login sessions and user stats after the session ends. Check the effect on application security by deleting the cookies. (I will soon write a separate article on cookie testing as well)
- **Validate your HTML/CSS:**
  - If you are optimizing your site for Search engines then HTML/CSS validation is the most important one.
  - Mainly validate the site for HTML syntax errors.
  - Check if the site is crawlable to different search engines.
- **Database Testing:**
  - Data consistency is also very important in a web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality.
  - Check if all the database queries are executing correctly, data is retrieved and also updated correctly. More on database testing could be a load on DB, we will address this in web load or performance testing below.

Functionality Testing of a Website

- **In testing the functionality of the websites the following should be tested:**
- **Links**
  i. Internal Links
  ii. External Links
  iii. Mail Links
  iv. Broken Links
- **Forms**
  i. Field validation
  ii. Error message for wrong input
  iii. Optional and Mandatory fields
- **Database**
  Testing will be done on database integrity.

Usability Testing

- Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.
  - Ease of learning
  - Navigation
  - Subjective user satisfaction
  - General appearance
- **Test for Navigation:**
  - Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.
- **Usability Testing includes the following:**
  - The website should be easy to use.
  - The instructions provided should be very clear.
  - Check if the instructions provided are perfect to satisfy its purpose.
  - The main menu should be provided on each page.
  - It should be consistent enough.

Usability Testing

- **Content Checking:**
  - Content should be logical and easy to understand. Check for spelling errors. The usage of dark colors annoys the users and should not be used in the site theme.
  - You can follow some standard colors that are used for web pages and content building. These are the commonly accepted standards like what I mentioned above about annoying colors, fonts, frames, etc.
  - Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.
  - These are some of the basic important standards that should be followed in web development. Your task is to validate all for UI testing.
- **Other user information for user help:**
  - Like the search option, the sitemap also helps files, etc. The sitemap should be present with all the links on websites with a proper tree view of navigation. Check for all links on the sitemap.
  - “Search in the site” option will help users to find content pages that they are looking for easily and quickly. These are all optional items and if present they should be validated.

Interface Testing

- In web testing, the server-side interface should be tested. This is done by verifying that communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.
- **The main interfaces are:**
  - Web server and application server interface
  - Application server and Database server interface.
  - Check if all the interactions between these servers are executed and errors are handled properly. If the database or web server returns an error message for any query by application server then the application server should catch and display these error messages appropriately to the users.
  - Check what happens if the user interrupts any transaction in-between? Check what happens if the connection to the webserver is reset in between?

Compatibility Testing

- The compatibility of your website is a very important testing aspect. See which compatibility test to be executed:
  - Browser compatibility
  - Operating system compatibility
  - Mobile browsing
  - Printing options
- **Browser Compatibility:** Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with.
- Your website coding should be a cross-browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application.
- Test web application on different browsers like Internet Explorer, Firefox, Netscape Navigator, AOL, Safari, Opera browsers with different versions.

Compatibility Testing

- **OS Compatibility:**
- Some functionality in your web application is that it may not be compatible with all operating systems. All new technologies used in web development like graphic designs, interface calls like different API's may not be available in all Operating Systems.
- Hence test your web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.
- **Mobile Browsing:**
- We are in the new technology era. So in future Mobile browsing will rock. Test your web pages on mobile browsers. Compatibility issues may be there on mobile devices as well.
- **Printing Options:**
- If you are giving page-printing options then make sure fonts, page alignment, page graphics, etc., are getting printed properly. Pages should fit the paper size or as per the size mentioned in the printing option.

Performance Testing

- The web application should sustain to heavy load. Web performance testing should include: Web Load Testing and Web Stress Testing
- Test application performance on different internet connection speeds.
- **Web Load Testing**: You need to test if many users are accessing or requesting the same page. Can the system sustain in peak load times? The site should handle many simultaneous user requests, large input data from users, simultaneous connection to DB, heavy load on specific pages, etc.
- **Web Stress Testing:** Generally stress means stretching the system beyond its specified limits. Web stress testing is performed to break the site by giving stress and it's checked as to how the system reacts to stress and how it recovers from crashes. Stress is generally given on input fields, login and sign up areas.
- In web performance, testing website functionality on different operating systems and different hardware platforms is checked for software and hardware memory leakage errors.
- Performance testing can be applied to understand the web site's scalability or to benchmark the performance in the environment of third-party products such as servers and middleware for potential purchase.

Performance Testing

- **Connection Speed**
  Tested on various networks like Dial-Up, ISDN, etc.
- **Load**
  i. What is the no. of users per time?
  ii. Check for peak loads and how the system behaves
  iii. A large amount of data accessed by the user
- **Stress**
  i. Continuous Load
  ii. Performance of memory, CPU, file handling, etc..

Security Testing

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing URL options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the URL site ID parameter to different site ID which is not related to the logged-in user. Access should be denied for this user to view other's stats.
- Try some invalid inputs in input fields like login username, password, input text boxes, etc. Check the system's reaction to all invalid inputs.
- Web directories or files should not be accessible directly unless they are given download option.
- Test the CAPTCHA for automating script logins.
- Test if SSL is used for security measures. If it is used, the proper message should get displayed when users switch from non-secure HTTP:// pages to secure HTTPS:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on the webserver.

Types Of Web Testing

- A website is classified into many types. Few of them are
    - Simple static website testing
    - Dynamic web application testing
    - E-commerce website testing
    - Mobile website testing

Simple Static Website

- A simple static website will display the same content for all visitors who are visiting the website at different times. It is also known as an informational website. In a static website, the only developer can do changes that too in code only. This type of website will not have any major functionalities and it purely depends on UI design.
- Testing a simple static website is very easy, you have to consider only a few things while testing.
- **#1)**Testing the GUI design is a must because a static website purely depends on it. You need to compare the approved PSD files with web page developed. Check all the elements in the design should present on the developed page.
- **#2)** The other part of GUI design is to check the font size, font style, spacing, and color everything has been reproduced.
- **#3)** Secondly, you need to check the links (page links) whether it is properly working or not? And also find, is there any broken link?
- **#4)** Verify the spelling and content in all web pages by comparing the content given by the client.

Simple Static Website

- **#5)** In some cases image will not display properly, it may break or sometimes images gets duplicated, wrong images may display. It has to be checked keenly. Because for a static website, only content and images will give lives.
- **#6)** Check the scroll bar carefully, in my experience, I have faced issues with the scrollbar. The issue you will face is unwanted scrolling appears or scroll gets hidden (it may hide the contents). The above issues are applicable to both horizontal and vertical scrolls.
- **#7)** If there is a contact form check it is working properly by sending some dummy messages.
- **#8)** Check whether it is an error-free web page, validate it with W3 validator or other related software.
- **#9)** Some constant things to be checked in a static website,
- Check favicon is present on the tab bar
- URL should contain the correct page title
- If copyright information is there, it should be displayed
- If there is a contact form, Captcha is a must. [It prevents junk email]
- Check the loading speed of the website. [A static website should not take much time for loading]. If a gif image is used while loading then track its functionality

Dynamic Web Application

- It is the type where the user can update and change their website content regularly.
- In addition to static website testing points –
- **#1)** In GUI section, ***tooltip is compulsory*** for all fields and buttons, field alignment (spacing) should be done properly, disabled field/ buttons should be greyed out, fields/ buttons should be in standard format as in SRS, error message should be displayed if something goes wrong, pop-up message should only display at the center of the web page, drop-down menu should not be truncated.
- Tab shortcut key should work in all fields and more.
- **#2)** In functionality section, if your web application is having login or sign up functionality then check the ***mandatory field validation***, form validation (i.e. number fields should accept only numbers, not alphabets), character restriction on fields (i.e. only these many characters can be entered).
- Special characters and negative numbers restriction on fields, testing the email functionality, testing the document upload (i.e. only ***specified document type can be uploaded***), timeout functionality, sorting functionality, javascript is working on compatible browsers etc should be tested.
- **#3)** When coming to back-end functionality section, test image uploading for broken images, text entering in the fields is working or not. Back-end update should ***reflect on front-end***, ***database testing*** (i.e. whether you can add new fields or deleting unwanted fields) all these things are to be performed.
- Performance is not much necessary for a web application (dynamic website) since it has very less content. If you need you can do with the tools with which you are familiar. Pick-up some standard online performance tool, if you want to do simple performance testing.

E-commerce Website

- An e-commerce website is somewhat complicated when compared to the above two. The tester needs to be very cautious while testing an e-commerce site. There are huge things to be checked in e-commerce sites out of them I just cover some of my experienced issues on e-commerce website testing.
- In the GUI section, you need to check all the features as in SRS and the same with the functionality. The functionality will be almost the same for all commercial websites.
- Functionality-wise you need to check all pages such as the main page (includes featured products, special offers display, log in details, search functionality) product detail page, category page, placing an order, payment gateway everything has to be tested.

E-commerce Website

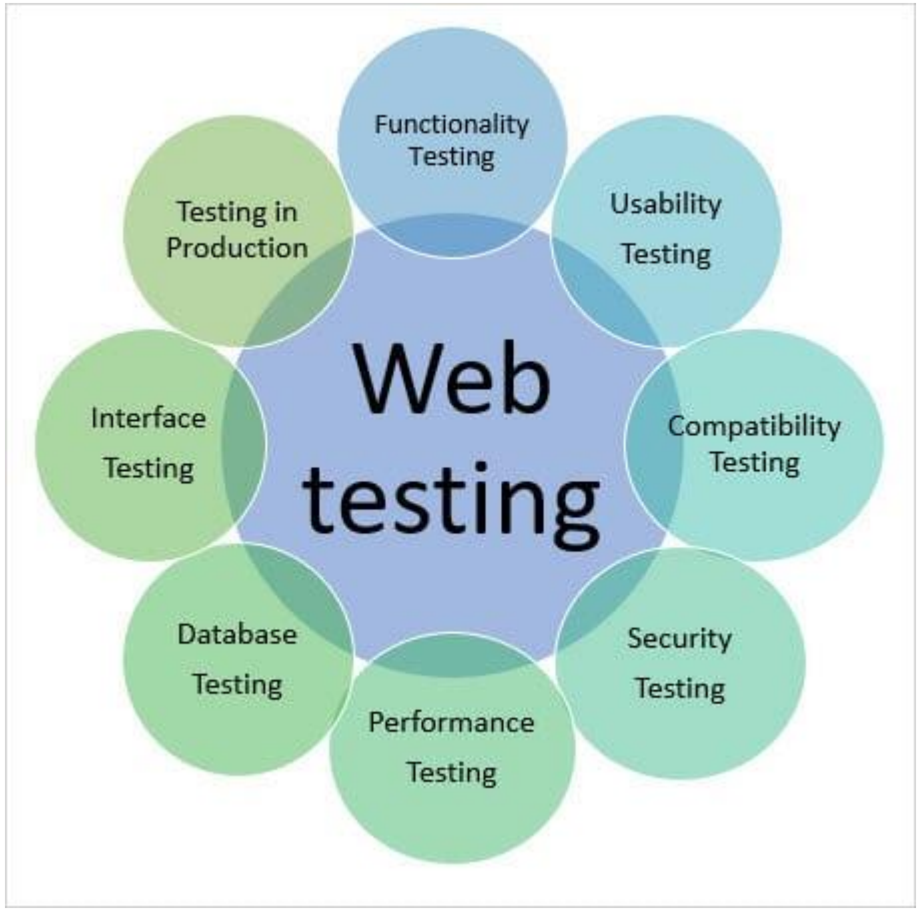
- **#1)** Check if the shopping cart is getting updated when you buy or increase the quantity. Check this functionality in all the pages and circumstances.
- **#2)** Check if special coupons and *offers are applied to correct orders* and you see the discounted price is displaying or not.
- **#3)** Sometimes while updating a single product it will get multiplied by considering the number of variations in the product. So check whether the single product is displayed and its variations are displayed correctly. (I faced this problem)
- **#4)** Check if the filter option whether is working exactly. If filtering is been done, based on the category & pricing chosen?
- **#5)** While sign up, super validation should be done. Only the new user can sign up.
- **#6)** If an existing user, added a product to the shopping basket, the wishlist section during their previous login should be saved and displayed during the next login too.
- **#7)** Compare products should work by comparing the products based on some specifications assigned in the back-end.
- **#8)** Check whether the Currency converter is working fine. Based on the country chosen the currency converter should display the relevant price and tax rates.

E-commerce Website

- **#9)** Generally many Plug-ins are used in an e-commerce (WordPress & similar) website, you need to be very careful. The plug-in installation may conflict or affect any other major functionality. So follow up with the plug-ins installation and its usage.
- **#10)** Check whether the social sharing option is working on the individual product or not.
- **#11)** Shipping cost should be generated based on the region selected. And also check the tax rate generation. (It may cause some legal problems, during the end-users purchase).
- **#12)** Payment gateway should work only if valid card details are given. Validation should apply to the Card number and CCV code number. [It is better to keep validation on the card number field itself].
- **#13)** Email generation on each and every process during purchase should happen (sign up, product ordering, payment successful, order canceled, order received and other email triggers if any).
- **#14)** Check the live chat with some dumpy emails.

Mobile Website

- A responsive website means making the content fit into the mobile device size instead of creating a version whereas a mobile website is creating a new version that is not a reflection desktop version. In the mobile website, you will have only limited pages, and unwanted functionalities are removed here.
- Testing a mobile website is somewhat tedious rather than other types of websites. It will have separate designs and you need to be careful while testing the functionalities.
- Usually, we will use an emulator for testing a mobile website and we can get ideal results but I always prefer you to test on real devices. I have faced many issues when I tested in real devices [Especially apple devices]. Real device specifications may conflict with the web pages developed.
- GUI & usability testing are more important as it is not the reflection of the desktop version.
- Performance is another important factor to be considered for mobile website testing. Performance-related issues can be tracked when you test in the real devices.
- Check whether browsing normal web links from mobile is getting triggered by a mobile link.
- Check page scrolling, page navigation, text truncation, etc on the mobile website.



Web testing
Functionality Testing
Usability Testing
Compatibility Testing
Security Testing
Performance Testing
Database Testing
Interface Testing
Testing in Production

Database Testing

- **DATABASE TESTING** is a type of software testing that checks the schema, tables, triggers, etc. of the Database under test. It also checks data integrity and consistency. It may involve creating complex queries to load/stress test the Database and check its responsiveness.
- **Why Database Testing**
- **Database Testing is Important** in software testing because it ensures data values and information received and stored into database are valid or not. Database testing helps to save data loss, saves aborted transaction data and no unauthorized access to the information. Database is important for any software application hence testers must have good knowledge of SQL for database testing.

Types of database testing

- Structural Testing
- Functional Testing
- Non-functional Testing

Structural Database Testing

- **Structural Database Testing** is a database testing technique that validates all the elements inside data repository that are mainly used for data storage and which are not allowed to be directly manipulated by end-users. The validation of database servers is also an important consideration in structural database testing. A successful completion of this testing needs mastery in SQL queries.
- **Schema Testing** in database testing validates various schema formats associated with the database and verifies whether the mapping formats of tables/views/columns are compatible with mapping formats of user interface. The main purpose of schema testing is to ensure the schema mapping between front-end and back-end are similar. Thus, it is also referred to as **mapping testing.**
- The most important **checkpoints** for schema testing -

- o Validation of the various schema formats associated with the databases. Many times the mapping format of the table may not be compatible with the mapping format present in the user interface level of the application.
- o There is a need for verification in the case of unmapped tables/views/columns.
- o There is also a need to verify whether heterogeneous databases in an environment are consistent with the overall application mapping.

Structural Database Testing

- Some of the interesting tools for validating database schemas.
- DBUnit that is integrated with Ant is very suitable for mapping testing.
- SQL Server allows the testers to be able to check and to query the schema of the Database by writing simple queries and not through code.
- For example, if the developers want to change a table structure or delete it, the tester would want to ensure that all the Stored Procedures and Views that use that table are compatible with the particular change.

Structural Database Testing

- various checks for database and column testing.
  - Whether the mapping of the database fields and columns in the backend is compatible with those mappings in the front-end?
  - Validation of the length and naming convention of the database fields and columns as specified by the requirements.
  - Validation of the presence of any unused/unmapped database tables/columns.
  - Validation of the compatibility of the data type, field lengths of the back-end database columns with that of those present at the front-end of the application.
  - Whether the database fields allow the user to provide desired user inputs as required by the business requirement specification documents.

Structural Database Testing

Important checks for keys and indexes -

- Check whether the required
  - Primary Key
  - Foreign Key
- constraints have been created on the required tables.
- Check whether the references for foreign keys are valid.
- Check whether the data type of the primary key and the corresponding foreign keys are the same in the two tables.
- Check whether the required naming conventions have been followed for all the keys and indexes.
- Check the size and length of the required fields and indexes.
- Whether the required Clustered indexes, Non Clustered indexes have been created on the required tables as specified by the business requirements.

Structural Database Testing

- **Stored procedures testing**
- Important tests to check stored procedures are:
  - Whether the development team did adopt the required
  - coding standard conventions
  - exception and error handling
  - for all the stored procedures for all the modules for the application under test.
  - Whether the development team did cover all the conditions/loops by applying the required input data to the application under test?
  - Whether the development team did properly apply the TRIM operation whenever data is fetched from the required tables in the Database?
  - Whether the manual execution of the Stored Procedure provides the end-user with the required result?
  - Whether the manual execution of the Stored Procedure ensures the table fields are being updated as required by the application under test?
  - Whether the execution of the Stored Procedures enables the implicit invoking of the required triggers?
  - Validation of the presence of any unused stored procedures.
  - Validation for Allow Null condition which can be done at the database level.
  - Validation of the fact that all the Stored Procedures and Functions have been successfully executed when the Database under test is blank.
  - Validation of the overall integration of the stored procedure modules as per as the requirements of the application under test.
  - Some of the useful tools for testing stored procedures are LINQ , SP Test tool etc.

Structural Database Testing

- **Trigger testing**
- Whether the required coding conventions have been followed during the coding phase of the Triggers?
- Check whether the triggers executed for the respective DML transactions have fulfilled the required conditions.
- Whether the trigger updates the data correctly once they have been executed?
- Validation of the required Update/Insert/Delete triggers functionality in the realm of the application under test.
- **Database server validations**
- Check the database server configurations as specified by the business requirements.
- Check the authorization of the required user to perform only those levels of actions that are required by the application.
- Check that the database server is able to cater to the needs of the maximum allowed number of user-transactions as specified by the business requirement specifications.

Functional Database Testing

- **Functional Database Testing** is a type of database testing that is used to validate the functional requirements of a database from the end-user’s perspective. The main goal of functional database testing is to test whether the transactions and operations performed by the end-users which are related to the database works as expected or not.
- Following are the basic conditions that need to be observed for database validations.
- Whether the field is mandatory while allowing NULL values on that field?
- Whether the length of each field is of sufficient size?
- Whether all similar fields have the same names across tables?
- Whether there are any computed fields present in the Database?

Functional Database Testing

- **Checking data integrity and consistency**
- Following checks are important
- Whether the data is logically well organized?
- Whether the data stored in the tables is correct and as per the business requirements?
- Whether there are any unnecessary data present in the application under test?
- Whether the data has been stored as per as the requirement with respect to data which has been updated from the user interface?
- Whether the TRIM operations performed on the data before inserting data into the Database under test?
- Whether the transactions have been performed according to the business requirement specifications and whether the results are correct or not?
- Whether the data has been properly committed if the transaction has been successfully executed?
- Whether the data has been rolled backed successfully if the transaction has not been executed successfully by the end-user?
- Whether the data has been rolled backed if the transaction has not been executed successfully and multiple heterogeneous databases have been involved in the transaction in question?
- Whether all the transactions have been executed by using the required design procedures as specified by the system business requirements?

Functional Database Testing

- **Login and user security**
- The validations of the login and user security credentials need to take into consideration the following things.
- Whether the application prevents the user from proceeding further in the application in case of a
- invalid username but valid password
- valid username but invalid password.
- invalid username and invalid password.
- Whether the user is allowed to perform only those specific operations which are specified by the business requirements?
- Whether the data is secured from unauthorized access?
- Whether there are different user roles created with different permissions?
- Whether all the users have required levels of access on the specified Database as required by the business specifications?
- Check that sensitive data like passwords, credit card numbers are encrypted and not stored as plain text in Database. It is a good practice to ensure all accounts should have passwords that are complex and not easily guessed.

Non-functional testing

- Non-functional testing in the context of database testing can be categorized into various categories as required by the business requirements.
- These can be load testing, Stress Testing, Security Testing, Usability Testing, and Compatibility Testing, and so on.
- The load testing, as well as stress testing, which can be grouped under the gamut of Performance Testing serves two specific purposes when it comes to the role of non-functional testing.

Non-functional testing

- **Load testing**
- The purpose of any load test should be clearly understood and documented. The following types of configurations are a must for load testing.
- The most frequently used user transactions have the potential to impact the performance of all of the other transactions if they are not efficient.
- At least one non-editing user transaction should be included in the final test suite, so that performance of such transactions can be differentiated from other more complex transactions.
- The more important transactions that facilitate the core objectives of the system should be included, as failure under a load of these transactions has, by definition, the greatest impact.
- At least one editable transaction should be included so that performance of such transactions can be differentiated from other transactions.
- Optimum response time under huge number of virtual users for all the prospective requirements.
- Effective times for fetching of various records.
- Important load testing tools are load runner, win runner and JMeter.

Non-functional testing

- **Database Stress Testing** is a testing method used to stress test database system with heavy load such that it fails at some point. This helps in identifying the breakdown point of database system.
- It requires proper planning and efforts in order to avoid over usage of resources. Data stress testing is also known as torturous testing or fatigue testing.
- Important stress testing tools are LoadRunner and JMeter.