

ASSIGNMENT MODULE - 2

1. What is Exploratory Testing?

Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously. Testing is often not recorded. Makes use of experience, heuristics and test patterns. Testing is based on a test charter that may include Scope of the testing (in and out). The focus of exploratory testing is more on testing as a “thinking” activity.

2. What is traceability matrix?

To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence. A software process should help you keeping the virtual table up-to-date. Simple technique may be quite valuable (naming convention).

3. What is Boundary value testing?

Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges. Boundary value analysis is a method which refines equivalence partitioning. Boundary value analysis generates test cases that highlight errors better than equivalence partitioning. The trick is to concentrate software testing efforts at the extreme ends of the equivalence classes. At those points when input values change from valid to invalid errors are most likely to occur. Boundary Value Analysis (BVA) uses the same analysis of partitions as EP and is usually used in conjunction with EP in test case design.

4. What is Equivalence partitioning testing?

Aim is to treat groups of inputs as equivalent and to select one representative input to test them all. EP can be used for all Levels of Testing. Equivalence partitioning is the process of defining the optimum number of tests by: Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function, Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition. If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g. value ≥ 1 and value ≤ 100) Then...”

5. What is Integration testing?

Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems. Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. Integration testing is done by a specific integration tester or test team. Components may be code modules, operating systems, hardware and even complete systems.

6. What determines the level of risk?

A Risk could be any future event with a negative consequence. You need to identify the risks associated with your project. Risks are of two types: Project Risks and Product Risk.

Example of Project risk is Senior Team Member leaving the project abruptly. Every risk is assigned likelihood i.e. chance of it occurring, typically on a scale of 1 to 10. Also the impact of that risk is identified on a scale of 1- 10. But just identifying the risk is not enough. You need to identify mitigation. In this case mitigation could be Knowledge Transfer to other team members & having a buffer tester in place

Example of product risks would be Flight Reservation system not installing in test environment Mitigation in this case would be conducting a smoke or sanity testing. Accordingly you will make changes in your scope items to include sanity testing.

7. What is Alpha testing?

It is always performed by the developers at the software development site. Sometimes it is also performed by Independent Testing Team. Alpha Testing is not open to the market and public it is conducted for the software application and project. It is always performed in Virtual Environment. It is always performed within the organization. It is the form of Acceptance Testing. Alpha Testing is definitely performed and carried out at the developing organizations location with the involvement of developers. It comes under the category of both White Box Testing and Black Box Testing.

8. What is beta testing?

It is always performed by the customers at their own site. It is not performed by Independent Testing Team. Beta Testing is always open to the market and public. It is usually conducted for software product. It is performed in Real Time Environment. It is always performed outside the organization. It is also the form of Acceptance Testing. Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data. It is only a kind of Black Box Testing.

9. What is component testing?

Component (Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.” Component Testing – The testing of individual software components. Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. Unit testing is the first level of testing and is performed prior to Integration Testing. Sometimes known as Unit Testing, Module Testing or Program Testing Component can be tested in isolation – stubs/drivers may be employed. Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in unit testing. Functional and Non-Functional testing Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool. A unit is the smallest testable part of an application like functions/procedures, classes, interfaces. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits. Unit tests find problems early in the development cycle. Unit testing is performed by using the White Box Testing method.

10. What is functional system testing?

A requirement that specifies a function that a system or system component must perform A Requirement may exist as a text document and/or a model.

11. What is Non-Functional Testing?

Non-Functional Testing: Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability May be performed at all Test levels (not just Non Functional Systems Testing) Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling Non-functional testing includes, but is not limited

to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing. It is the testing of “how” the system works. Non-functional testing may be performed at all test levels. The term non-functional testing describes the tests required to measure characteristics of systems and software that can be quantified on a varying scale, such as response times for performance testing. To address this issue, performance testing is carried out to check & fine tune system response times. The goal of performance testing is to reduce response time to an acceptable level hence load testing is carried out to check systems performance at different loads i.e. number of users accessing the system

12. What is GUI Testing?

Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc

13. What is Adhoc testing?

Error guessing is a technique where experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

Some people seem to be naturally good at testing and others are good testers because they have a lot of experience either as a tester or working with a particular system and so are able to find out its weaknesses.

This is why an error guessing approach, used after more formal techniques have been applied to some extent, can be very effective. It also saves a lot of time because of the assumptions and guessing made by the experienced testers to find out the defects which otherwise won’t be able to find. Using experience to postulate errors. Use Error Guessing to Complement Test Design Techniques.

14. What is load testing?

Some extremely popular sites have suffered serious downtimes when they get massive traffic volumes. E-commerce websites invest heavily in advertising campaigns, but not in Load Testing to ensure optimal system performance, when that marketing brings in traffic.

15. What is stress Testing?

Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant.

16. What is white box testing and list the types of white box testing?

Testing based on an analysis of the internal structure of the component or system.

Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

Types of white box testing

1. Test/Code Coverage
2. Statement/Segment Coverage
3. Decision/Branch Coverage
4. Condition Coverage

17. What is black box testing? What are the different black box testing techniques?

Testing, either functional or non-functional, without reference to the internal structure of the component or system. Specification-based testing technique is also known as 'black-box' or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

There are four specification-based or black-box techniques:

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing
- Use-case Testing
- Other Black Box Testing
- Syntax or Pattern Testing

18. Mention what are the categories of defects?

- **Data Quality/Database Defects:** Deals with improper handling of data in the database.
- Examples:
 - Values not deleted/inserted into the database properly
 - Improper/wrong/null values inserted in place of the actual values
 - **Critical Functionality Defects:** The occurrence of these bugs hampers the crucial functionality of the application. Examples: - Exceptions
 - **Functionality Defects:** These defects affect the functionality of the application.
- Examples:
 - All JavaScript errors
 - Buttons like Save, Delete, Cancel not performing their intended functions
 - A missing functionality (or) a feature not functioning the way it is intended to
 - Continuous execution of loops
 - **Security Defects:** Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
- Examples:
 - Authentication: Accepting an invalid username/password
 - Authorization: Accessibility to pages though permission not given

- **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

- Examples:

- improper error/warning/UI messages

- spelling mistakes

- Alignment problems

19. Mention what big bang testing is?

In Big Bang integration testing all components or modules are integrated simultaneously, after which everything is tested as a whole.

Big Bang testing has the advantage that everything is finished before integration testing starts.

20. What is the purpose of exit criteria?

Successful Testing of Integrated Application.

Executed Test Cases are documented

All High prioritized bugs fixed and closed

Technical documents to be submitted followed by release Notes.

21. When should "Regression Testing" be performed?

If your software undergoes frequent changes, regression testing costs will escalate.

In such cases, Manual execution of test cases increases test execution time as well as costs.

Automation of regression test cases is the smart choice in such cases.

Extent of automation depends on the number of test cases that remain re-usable for successive regression cycles.

22. What are 7 key principles? Explain in detail?

1. Testing shows presence of Defects

Testing can show that defects are present, but cannot prove that there are no defects.

- Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness.

- We test to find Faults

- As we find more defects, the probability of undiscovered defects remaining in a system reduces.

- However Testing cannot prove that there are no defects present

2. Exhaustive Testing is Impossible!

Testing everything including all combinations of inputs and preconditions is not possible.

- so, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
- For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.
- This is very unlikely that the project timescales would allow for this number of tests.
- So, accessing and managing risk is one of the most important activities and reason for testing in any project.
- We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions).
- That is we must Priorities our testing effort using a Risk Based Approach.

3. Early Testing

Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.

- Testing activities should start as early as possible in the development life cycle
- These activities should be focused on defined objectives – outlined in the Test Strategy
- Remember from our Definition of Testing, that Testing doesn't start once the code has been written!

4. Defect Clustering

A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.

- Defects are not evenly spread in a system
- They are 'clustered'
- In other words, most defects found during testing are usually confined to a small number of modules
- Similarly, most operational failures of a system are usually confined to a small number of modules
- An important consideration in test prioritization!

5. The Pesticide Paradox

If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.

- To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need

to be written to exercise different parts of the software or system to potentially find more defects.

- Testing identifies bugs, and programmers respond to fix them
- As bugs are eliminated by the programmers, the software improves
- As software improves the effectiveness of previous tests erodes

6. Testing is Context Dependent

Therefore we must learn, create and use new tests based on new techniques to catch new bugs

● N.B It's called the "pesticide paradox" after the agricultural phenomenon, where bugs such as the boll weevil build up tolerance to pesticides, leaving you with the choice of ever-more powerful pesticides followed by ever-more powerful bugs or an altogether different approach.' – Beizer 1995

7. Absence of Errors Fallacy

If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.

- if we build a system and, in doing so, find and fix defects....
- It doesn't make it a good system
- Even after defects have been resolved it may still be unusable and/or does not fulfill the users' needs and expectations

23. Difference between QA v/s QC v/s Tester

| SR no | Quality Assurance | Quality Control | Testing |
|-------|---|--|--|
| 1 | Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements. | Activities which ensure the identification of bugs/errors/defects in the Software. |
| 2 | Focuses on processes and procedures rather than conducting actual testing on the system. | Focuses on actual testing by executing Software with intend to identify bugs/defects through implementation of procedures and processes. | Focuses on actual testing |
| 3 | Process oriented activities. | Product oriented activities. | Product oriented activities. |
| 4 | Preventive activities. | It is a corrective process. | It is a preventive process. |
| 5 | It is a subset of the Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance | Testing is the subset of Quality Control. |

24. Difference between Smoke and Sanity?

| Smoke Testing | Sanity Testing |
|---------------|----------------|
|---------------|----------------|

| | |
|---|--|
| Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine | Sanity Testing is done to check the new functionality / bugs have been fixed |
| The objective of this testing is to verify "stability" of the system in order to with more rigorous testing | The objective of the testing is to verify the "rationality" of the system in order proceed to proceed with more rigorous testing |
| This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| Smoke testing is usually documented. | Sanity testing is usually not documented and or scripted is unscripted |
| Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |
| Smoke testing exercises the entire system | Sanity testing exercises only the from end to end particular component of the entire system |
| Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

25. Difference between verification and Validation

Verification - The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.

Validation - The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.

26. Explain types of Performance testing.

Types of Performance Testing

- **Load is testing** - It's a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.
- **Stress testing** - System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
- **Endurance testing** - Stress testing is also known as endurance testing.
- **Spike testing** - Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load. The goal of spike testing is to determine the behavior of a software application when it receives extreme variations in traffic.
- **Volume testing** - Volume Testing is a type of software testing which is carried out to test a software application with a certain amount of data. The amount used in volume testing could be a database size or it could also be the size of an interface file that is the subject of volume testing.

● **Scalability testing** - A scalability test is a type of load testing that measures the application's ability to scale up or down as a reaction to an increase in the number of users. In other words, it tests how the system is going to perform during a sudden spike or fall of user request loads.

27. What is Error, Defect, Bug and failure?

A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then it is called bug, build does not meet the requirements then it is failure”

Error - A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

Defect - Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Bug - A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.

Failure - The inability of a system or component to perform its required functions within specified performance requirements. See: bug, crash, exception, and fault.

28. Difference between Priority and Severity

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

29. What is Bug Life Cycle?

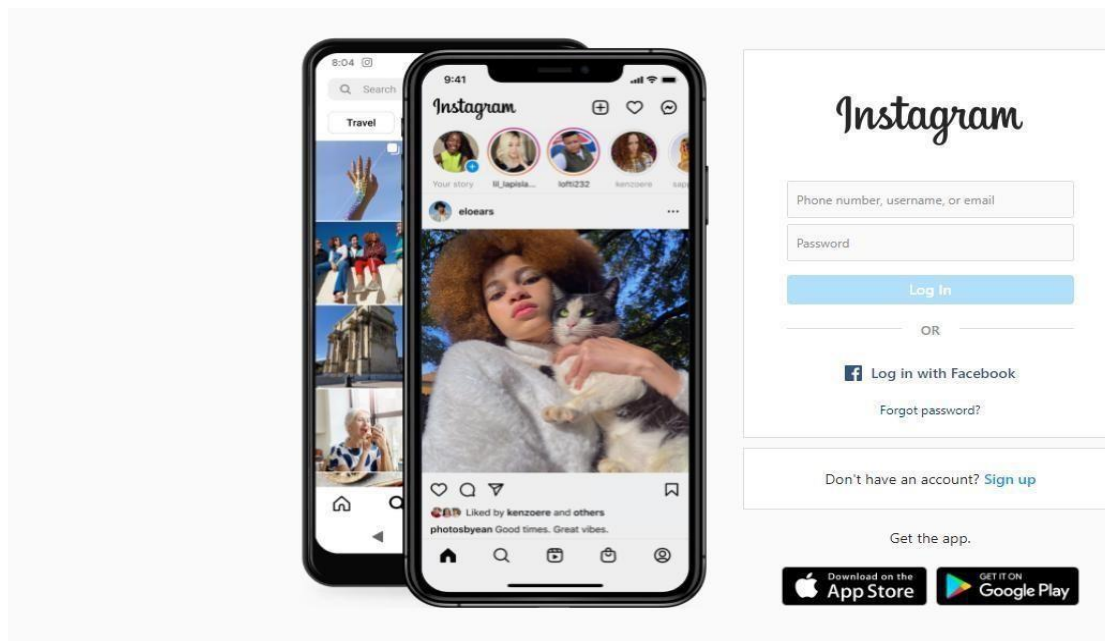
A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program's source code or its design.

30. Explain the difference between Functional testing and Nonfunctional testing

| Functional Testing | Non-Functional Testing |
|--|--|
| It is performed before non-functional testing. | It is performed after the functional testing. |
| It is based on customer's requirements. | It focuses on customer's expectation. |
| It is easy to define functional requirements. | It is difficult to define the requirements for non-functional testing. |
| Helps to validate the behavior of the application. | Helps to validate the behavior of the application. |

| | |
|--|--|
| Carried out to validate software actions. | It is done to validate the performance of the software. |
| Functional testing is carried out using the functional specification. | This kind of testing is carried out by performance specifications |
| Functional testing is easy to execute by manual testing. | It's very hard to perform non-functional testing manually. |
| It describes what the product does. | It describes how the product works. |
| Check login functionality. | Check login functionality. |
| <ul style="list-style-type: none"> ● Unit testing ● Smoke testing ● User Acceptance ● Integration Testing ● Regression testing ● Localization ● Globalization ● Interoperability | <ul style="list-style-type: none"> ● Performance Testing ● Volume Testing ● Scalability ● Usability Testing ● Load Testing ● Stress Testing ● Compliance Testing ● Portability Testing ● Disaster Recover Testing |

31. To create HLR & Test Case of 1)(Instagram , Facebook) only first page



| HLR | | |
|-----|----------------------------|---|
| ID | Functionality name | Functionality Description |
| | | |
| | Instagram URL | |
| 100 | Instagram URL | while entering URL it open the website |
| | Instagram Google | While search on Google it open the website. |
| | | |
| | Website logo | |
| 200 | Instagram logo | When I click on logo it doesn't responding |
| | | |
| | Login button | |
| 300 | Login button | When I click on login button it enters the home page of instagram. |
| | | |
| | Login with Facebook | |
| 400 | Login with Facebook | When I click on login with Facebook button, it enters the home page of instagram. |
| | | |
| | Forgot password | |
| 500 | Forgot password | When I click on forgot password, it redirects to the reset password page. |
| | | |
| | Sign up | |
| 600 | Sign up | When I click on sign up, it redirects to the sign up page |
| | | |
| | Google play | |
| 700 | Google play | When I click on get it on Google play, it redirects to the Google playstore page |
| | | |

NOTE: For Test Case Separate Excel sheet Attached Kindly refer it.

2) Facebook Login Page: <https://www.facebook.com/>



| HLR | | |
|-----|---------------------------|--|
| ID | Functionality name | Functionality Description |
| | | |
| | Facebook URL | |
| 100 | Facebook URL | while entering URL it open the website |
| | Facebook Google | While search on Google it open the website. |
| | | |
| | Website logo | |
| 200 | Facebook logo | When I click on logo it doesn't responding |
| | | |
| | Login button | |
| 300 | Login button | When I click on login button it enters the home page Facebook. |
| | | |
| | Forgotten Password | |
| 400 | Forgotten password | When I click on Forgotten password link, it enters the Find your account page. |
| | | |
| | Create New Account | |
| 500 | Create New Account | When I click on Create new account, it redirects to the sign up page. |
| | | |
| | Create a Page | |
| 600 | Create a Page | When I click on Create a page link, it redirects to the Create a page |
| | | |

NOTE: For Test Case Separate Excel sheet Attached Kindly refer it.

32. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

| SDLC | STLC |
|--|---|
| Development Life Cycle | Testing Life Cycle |
| The main object of SDLC life cycle is to complete successful development of the software including testing and other phases. | The only objective of the STLC phase is testing. |
| In SDLC the business analyst gathers the requirements and create Development Plan | In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan |
| In SDLC, the development team creates the high and low-level design plans | In STLC, the test analyst creates the Integration Test Plan |
| The real code is developed, and actual work takes place as per the design documents. | The testing team prepares the test environment and executes them |
| SDLC phase also includes post-deployment supports and updates. | Testers, execute regression suits, usually automation scripts to check maintenance code deployed. |

33. What is the difference between test scenarios, test cases, and test script?

| Test Scenario | Test Case | Test Script |
|---|--|---|
| Test scenario is any Functionality that can be tested. | Test case is a set of action executed to verify particular features or functionality. | Test script is a set of instructions to test an automatically. |
| It is derived from test artifacts like business requirement specification and software requirement specification. | It is mostly derived from test scenario. | It is mostly derived from test cases. |
| Helps test the end-to-end functionality in agile way. | Helps in exhaustive testing of an app. | Helps to test specific things repeatedly. |
| It is more focused on what to test. | It is focused on what to test and how to test. | It is focused on the expected result. |
| Takes less time and fewer resources to create. | Requires more resources and time. | Requires less time for testing but more resources for scripts creating and updating. |
| Includes an end-to-end functionality to be tested. | Includes test steps, data, expected result for testing. | Includes different commands to develop script. |
| The main task is to check the full functionality of a software application. | The main task is to verify compliance with the applicable standards, guidelines and customer requirements. | The main task is to verify that nothing is skipped and the results are true as the desired testing plan |
| Allows quickly assessing the testing scope | Allows detecting error and defects. | Allow carrying out an automatic execution of test cases. |

34. Explain what Test Plan is? What is the information that should be covered?

A document describing the scope, approach, resources and schedule of intended test activities. Determining the scope and risks, and identifying the objectives of testing. Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

35. What is priority?

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

For example: If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

Priority can be of following types:

Low: The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.

Medium: The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.

High: The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.

Critical: Extremely urgent, resolve immediately

36. What is severity?

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

Severity can be of following types:

Critical: The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable and there is no acceptable alternative method to achieve the required results then the severity will be stated as critical.

Severity can be of following types:

Major (High): The defect that results in the termination of the complete system or one or more component of the system and causes extensive corruption of the data. The failed function is unusable but there exists an acceptable alternative method to achieve the required results then the severity will be stated as major.

Moderate (Medium): The defect that does not result in the termination, but causes the system to produce incorrect, incomplete or inconsistent results then the severity will be stated as moderate.

Minor (Low): The defect that does not result in the termination and does not damage the usability of the system and the desired results can be easily obtained by working around the defects then the severity is stated as minor.

Cosmetic: The defect that is related to the enhancement of the system where the changes are related to the look and field of the application then the severity is stated as cosmetic.

37. Bug categories are...

New: When a new defect is logged and posted for the first time. It is assigned a status as NEW.

Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team.

Open: The developer starts analyzing and works on the defect fix

Fixed: When a developer makes a necessary code change and verifies the change, he or she can make bug status as "Fixed."

Pending retest: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”

Retest: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”

Verified: The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”

Reopen: If the bug persists even after the developer has fixed the bug, the tester changes the status to “reopened”. Once again the bug goes through the life cycle.

Closed: If the bug is no longer exists then tester assigns the status “Closed.”

Duplicate: If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to “duplicate.”

Rejected: If the developer feels the defect is not a genuine defect then it changes the defect to “rejected.”

Deferred: If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status “Deferred” is assigned to such bugs.

Not a bug: If it does not affect the functionality of the application then the status assigned to a bug is “Not a bug”.

38. Advantage of Bugzilla.

Bugzilla is an open-source issue/bug tracking system that allows developers to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database

39. Difference between priority and severity

| Priority | Severity |
|---|---|
| Defect Priority has defined the order in which the developer should resolve a defect | Defect Severity is defined as the degree of impact that a defect has on the operation of the product |
| Priority is associated with scheduling | Severity is associated with functionality or standards |
| Priority indicates how soon the bug should be fixed | Severity indicates the seriousness of the defect on the product functionality |
| Priority of defects is decided in consultation with the manager/client | QA engineer determines the severity level of the defect |
| Priority is driven by business value | Severity is driven by functionality |
| Its value is subjective and can change over a period of time depending on the change in the project situation | Its value is objective and less likely to change |
| High priority and low severity status indicates, defect have to be fixed on immediate bases but does not affect the application | High severity and low priority status indicates defect have to be fixed but not on immediate bases |
| Priority status is based on customer requirements | Severity status is based on the technical aspect of the product |
| During UAT the development team fix defects based on priority | During SIT, the development team will fix defects based on the severity and then priority |
| Priority is categorized into three types <ul style="list-style-type: none">• Low• Medium• High | Severity is categorized into five types <ul style="list-style-type: none">• Critical• Major• Moderate• Minor |

- Cosmetic

40. What are the different Methodologies in Agile Development Model?

The **Agile methodology** is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the **work** begins, teams cycle through a process of planning, executing, and evaluating.

41. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

| Authentication | Authorization |
|---|--|
| Authentication verifies who the user is. | Authorization determines what resources a user can access. |
| Authentication works through <u>passwords</u> , one-time pins, biometric information, and other information provided or entered by the user. | Authorization works through settings that are implemented and maintained by the organization. |
| Authentication is the first step of a good identity and access management process. | Authorization always takes place after authentication. |
| Authentication is visible to and partially changeable by the user. | Authorization isn't visible to or changeable by the user. |
| Example: By verifying their identity, employees can gain access to a human resources (HR) application that includes their personal pay information, vacation time, and 401K data. | Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization. |

Integration testing exposes problems with interfaces among different program components before deployment.

Interoperability. ...

Security. ...

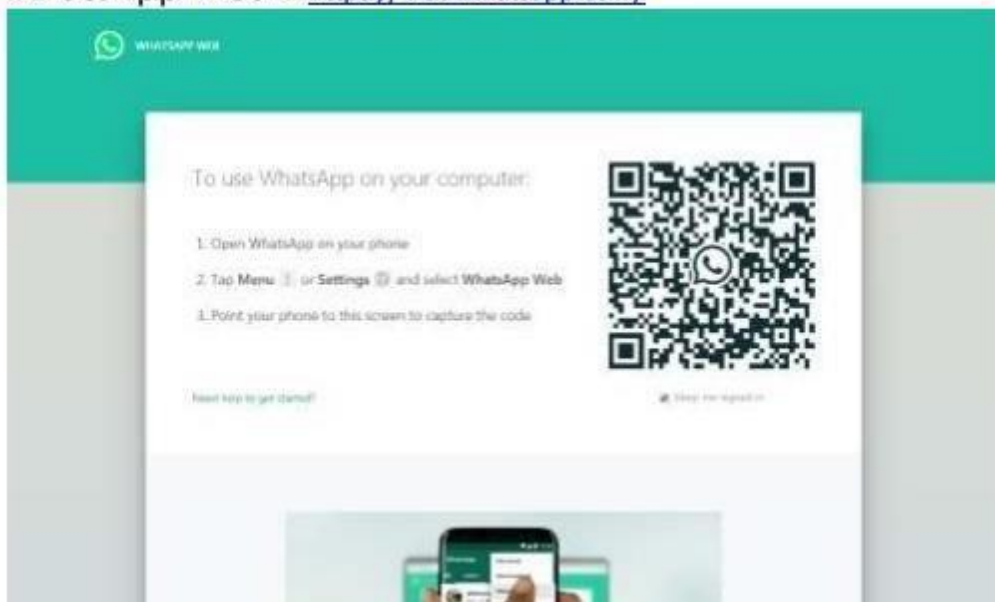
Performance. ...

Usability. ...

Quality Testing, Exceptional Services.

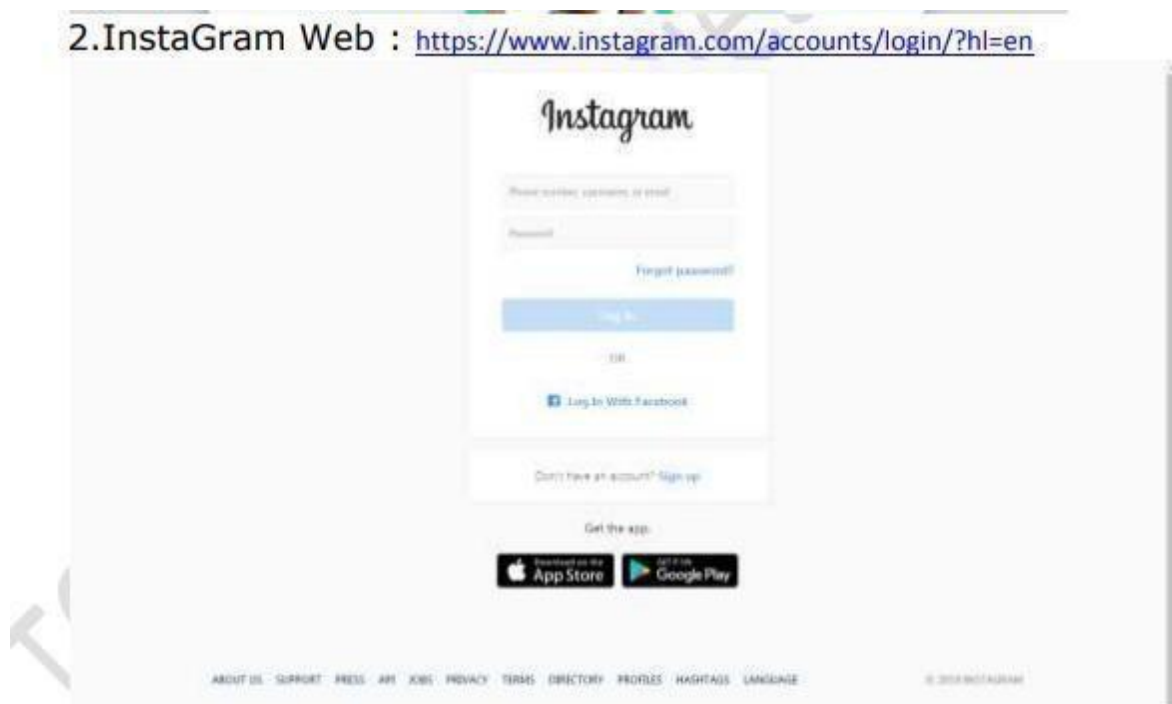
42. To create HLR & Test Case of Web Based (WhatsApp web, Instagram) 1. WhatsApp Web : <https://web.whatsapp.com/>

1. WhatsApp Web : <https://web.whatsapp.com/>



| HLR | | |
|-----|---------------------------------|--|
| ID | Functionality name | Functionality Description |
| | | |
| | Whatsapp Web URL | |
| 100 | Whatsapp Web URL | while entering URL it open the website |
| | Whatsapp Web Google | While search on Google it open the website. |
| | | |
| | Website logo | |
| 200 | Whatsapp logo | When I click on logo it doesn't responding |
| | | |
| | Link with phone number | |
| 300 | Link with Phone number | When I click on Link with Phone number, it redirects to the enter the phone number page. |
| | | |
| | Need help to get started | |
| 400 | Need help to get started | When I click on Need help to get started, it redirects to the Help Centre Page. |
| | | |
| | Play Button | |
| 500 | Play Button | When I click on Play button, It plays the video. |
| | | |

NOTE: For Test Case Separate Excel sheet Attached Kindly refer it.



| HLR | | |
|-----|--------------------|---------------------------|
| ID | Functionality name | Functionality Description |

| | | |
|-----|----------------------------|---|
| | | |
| | Instagram URL | |
| 100 | Instagram URL | while entering URL it open the website |
| | Instagram Google | While search on Google it open the website. |
| | | |
| | Website logo | |
| 200 | Instagram logo | When I click on logo it doesn't responding |
| | | |
| | Login button | |
| 300 | Login button | When I click on login button it enters the home page of instagram. |
| | | |
| | Login with Facebook | |
| 400 | Login with Facebook | When I click on login with Facebook button, it enters the home page of instagram. |
| | | |
| | Forgot password | |
| 500 | Forgot password | When I click on forgot password, it redirects to the reset password page. |
| | | |
| | Sign up | |
| 600 | Sign up | When I click on sign up, it redirects to the sign up page |
| | | |
| | Google play | |
| 700 | Google play | When I click on get it on Google play, it redirects to the Google play store page |
| | | |

NOTE: For Test Case Separate Excel sheet Attached Kindly refer it.

43. To create HLR and Test Case on this Link. <https://artoftesting.com/>

Keep in touch |

Reach us for query or concern regarding any post on ArtOfTesting, we will try our best to resolve your query.

Name

Email

Subject

Message

SEND MESSAGE

| HLR | | |
|-----|----------------------|--|
| ID | Functionality name | Functionality Description |
| | | |
| | Instagram URL | |
| 100 | artoftesting URL | while entering URL it open the website |

| | | |
|-----|---------------------|---|
| | Instagram Google | While search on Google it open the website. |
| | | |
| | Send message | |
| 200 | Send Message Button | When I click on Send Message button it enters the Contact Us page of artoftesting page. |

NOTE: For Test Case Separate Excel sheet Attached Kindly refer it.

44. Write a scenario of only Whatsapp chat messages

- Verify that the user can send messages to any individual selected from his contact list.
- Verify that 'Chats' window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.
- Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person.
- Verify that the user can check the message delivered and read the time for a message in the 'Message Info' section.
- Verify that the user can share or receive contact with the other person.
- Verify that the user can create a group by adding multiple people from his contact list.
- Verify that the user can send and receive the message in group chats.
- Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals.
- Verify that users can send and receive images, audio, video, and emoticons in group chats.
- Verify that the user can send and receive chats in the secondary languages available.
- Verify that users can delete text, images, audio, and video messages within a chat.
- Verify that users can clear their complete chat history in an individual or group chat.
- Verify that users can archive chats in an individual or group chat.
- Verify that users can block a user to prevent any message from getting received from the blocked contact.

45. Write a Scenario of Pen

- Verify that the length and the diameter of the pen are as per the specifications.
- Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen.
- Verify that the user is able to write clearly over different types of papers.
- Check the weight of the pen. It should be as per the specifications. In case not mentioned in the specifications, the weight should not be too heavy to impact its smooth operation.
- Verify if the pen is with a cap or without a cap.
- Check the color of the outer body of the pen. It should be as per the specifications.
- Verify the color of the ink on the pen.
- Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
- Verify if the text written by the pen is erasable or not.
- Verify the strength of the pen's outer body. It should not be easily breakable.
- Verify if the pen can support multiple refills or not.
- In the case of a ball and gel pen, verify that the user can change the refill of the pen easily.

46. Write a Scenario of Pen Stand

- Verify that the material is used for pen stand such as like wood, iron or any other.
- Verify the type of pen stand, whether it is a ballpoint pen, pencil, scale, or any other instrument.
- Verify that the pen stand is a with clock or single.

- Check the weight of the pen stand. The weight should not be too heavy to carry.
- Check the color of the outer body of the pen stand. It should be as per the specifications.
- Verify the surfaces over which the pen stand is able to put on table.

47. Write a Scenario of Door

- Verify if the door is single door or bi-folded door
- Verify that the dimension of the doors are as per the specifications
- Verify that the material used in the door body and its parts is as per the specifications
- Verify that color of the door is as specified
- Verify if the door is sliding door or rotating door
- Check the position, quality and strength of hinges
- Check the type of locks in the door
- Check the number of locks in the door interior side or exterior side
- Verify if the door is having peek-hole or not
- Verify if the door is having stopper or not
- Verify if the door closes automatically or not – spring mechanism
- Verify if the door makes noise when opened or closed
- Check the door condition when used extensively with water

48. Write a Scenario of ATM

- Verify that all the labels and controls including text boxes, buttons, images, and links are present on the screen.
- Check the informative text written displayed on the screen is clearly visible and legible.
- Verify that the size, color, and UI of the different objects are as per the specifications.
- Verify that the application's UI is responsive i.e. it should adjust to different screen resolutions of ATM machines.
- Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both.
- Verify that on properly inserting a valid card different banking options appear on the screen.
- Verify that the touch of the ATM screen is smooth and operational.
- Verify that the user is presented with the option to choose a language for further operations.
- Check that the user is asked to enter a pin number before displaying any card/bank account detail.
- Verify that there are a limited number of attempts up to which the user is allowed to enter the pin code.
- Check that the pin is displayed in masked form when entered.
- Verify that the user is allowed to get account details like available balance.
- Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications.
- Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.
- Verify that the user is provided the option to get the transaction details in printed form.
- Verify that the user's session timeout is maintained.
- Verify that the user is not allowed to exceed the one-day transaction limit amount.

- Verify that the user is allowed to do only one transaction per pin request.
- Verify that the applicable fee gets deducted along with the withdrawn amount in case the user exceeds the limit of the number of free transactions in a month.
- Verify that the applicable fee gets deducted along with the withdrawn amount in case the user uses a card of a bank other than that of an ATM.
- Verify that in case of sudden electricity loss before withdrawing cash, the transaction is marked as null and the amount is not withdrawn from the user's account.

49. When to use Usability Testing?

Aesthetics and design are important. How well a product looks usually determines how well it works. There are many software applications / websites, which miserably fail, once launched, due to following reasons

Where do I click next?

Which page needs to be navigated?

Which Icon or Jargon represents what?

Error messages are not consistent or effectively displayed

Session time not sufficient.

Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure.

50. What is the procedure for GUI Testing?

MANUAL BASED TESTING

Under this approach, graphical screens are checked manually by testers in conformance with the requirements stated in business requirements document.

RECORD AND REPLAY

GUI testing can be done using automation tools. This is done in 2 parts. During Record, test steps are captured into the automation tool. During playback, the recorded test steps are executed on the Application under Test. Example of such tools - QTP.

MODEL BASED TESTING

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior. Models help in a generation of efficient test cases using the system requirements.

51. Write a scenario of Microwave Oven

- Verify that the dimensions of the oven are as per the specification provided.
- Verify that the oven's material is optimal for its use as an oven and as per the specification.
- Verify that the oven heats the food at the desired temperature properly.
- Verify that oven heats food at the desired temperature within a specified time duration.
- Verify the ovens functioning with maximum attainable temperature.
- Verify the ovens functioning with minimum attainable temperature.
- Verify that the oven's plate rotation speed is optimal and not too high to spill the food kept over it.
- Verify that the oven's door gets closed properly.
- Verify that the oven's door opens smoothly.
- Verify the battery requirement of the microwave oven and check that it functions smoothly at that power.
- Verify that the text written over the oven's body is clearly readable.
- Verify that the digital display is clearly visible and functions correctly.
- Verify that the temperature regulator is smooth to operate.
- Verify that the temperature regulator works correctly.
- Check the maximum capacity of the oven and test its functioning with that volume of food.

- Check oven's functionality with different kinds of food – solid, liquid.
- Check the oven's functionality with different food at different temperatures.
- Verify the oven's functionality with different kinds of container material.
- Verify that the power cord of the oven is long enough.
- Verify that the usage instruction or user manuals have clear instructions.

52. Write a scenario of Coffee vending Machine

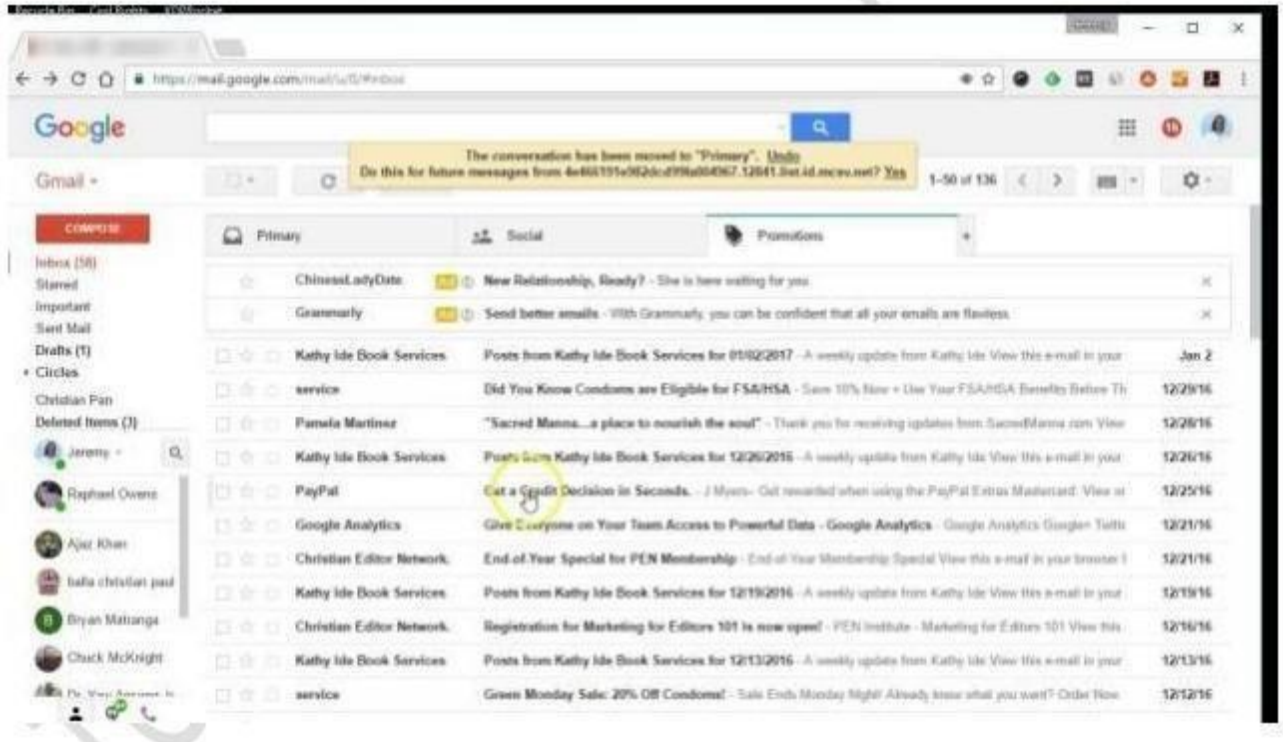
- Verify that the coffee vending machine should be started when the user presses the Power ON button.
- Verify that coffee vending machine should be off when the user press on power OFF button
- Verify that vending machine all buttons should be working properly
- Verify that when the vending machine starts, the indicator lights should be working properly.
- Verify that the mechanism should be working properly when ingredients are under capacity level
- Verify that the auto cleaner facility is working properly or not
- Verify that the water level indicator should be working properly.
- Verify that the half-cup facility is working properly or not
- Verify that the cup quantity counter should work properly.
- Verify that the automatic temperature is working properly or not
- Verify that the safety lock system is available or not.
- Verify that the cleaner should work properly for the coffee vending machine.

53. Write a scenario of chair

- Verify that the chair is stable enough to take an average human load.
- Check the material used in making the chair-wood, plastic etc.
- Check if the chair's leg are level to the floor
- Check the usability of the chair as an office chair, normal household chair
- Check if there is back support in the chair
- Check if there is support for hands in the chair
- Verify the paint's type and color
- Verify if the chair's material is brittle or not
- Check if cushion is provided with chair or not
- Check the condition when washed with water or effect of water on chair
- Verify that the dimension of chair is as per the specifications
- Verify that the weight of the chair is as per the specifications
- Check the height of the chair's seat from floor

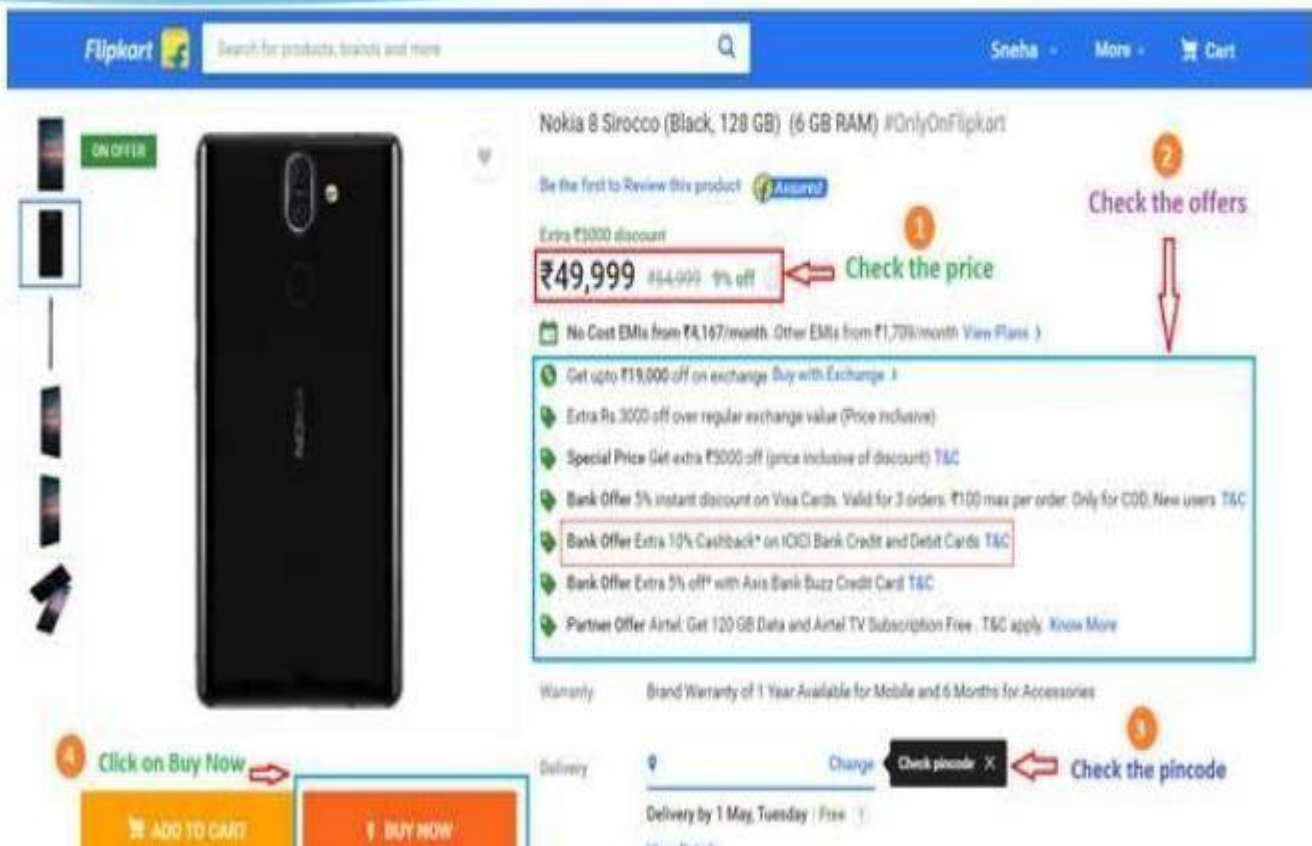
54. To Create Scenario (Positive & Negative)

2. Gmail (Receiving mail)



- Verify that a newly received email is displayed as highlighted in the Inbox section.
- Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body (trimmed to a single line).
- Verify that on clicking the newly received email, the user is navigated to email content.
- Verify that the email contents are correctly displayed with the desired source formatting.
- Verify that any attachments are attached to the email and are downloadable.
- Verify that the attachments are scanned for viruses before download.
- Verify that all the emails marked as read are not highlighted.
- Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.
- Verify that count of unread emails is displayed alongside 'Inbox' text in the left sidebar of Gmail.
- Verify that unread email count increases by one on receiving a new email.
- Verify that unread email count decreases by one on reading an email (marking an email as read).
- Verify that email recipients in cc are visible to all users.
- Verify that email recipients in bcc are not visible to the user.
- Verify that all received emails get piled up in the 'Inbox' section and get deleted in cyclic fashion based on the size availability.
- Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc.

Online shopping to buy product (flipkart)



- Verify that the price is correct or not of the product.
- Verify that the different offers are showing or not.
- Verify that the user can enter pincode or not.
- Verify that the user can click on buynow button or not.

55. Write a Scenario of Wrist Watch

- Verify the type of watch – analog or digital.
- In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.
- In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed.
- Verify the material of the watch and its strap.
- Check if the shape of the dial is as per specification.
- Verify the dimension of the watch is as per the specification.
- Verify the weight of the watch.
- Check if the watch is waterproof or not.
- Verify that the numbers in the dial are clearly visible or not.
- Check if the watch is having a date and day display or not.
- Verify the color of the text displayed in the watch – time, day, date, and other information.
- Verify that clock's time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.
- Check if the second hand of the watch makes ticking sound or not.
- Verify if the brand of the watch and check if it's visible in the dial.
- Check if the clock is having stopwatch, timers, and alarm functionality or not.
- In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.
- Verify if the watch comes with any guarantee or warranty.
- Verify if the dial has glass covering or plastic, check if the material is breakable or not.

- Verify if the dial's glass/plastic is resistant to minor scratches or not.
- Check the battery requirement of the watch.

56. Write a Scenario of Lift(Elevator)

- Verify the dimensions of the lift
- Verify the type of door of the lift is as per the specification
- Verify the type of metal used in the lift interior and exterior
- Verify the capacity of the lift in terms of the total weight
- Verify the buttons in the lift to close and open the door and numbers as per the number of floors
- Verify that lift moves to the particular floor as the button of the floor is clicked
- Verify that lift stops when up/down buttons at particular floor are pressed
- Verify if there is an emergency button to contact officials in case of any mishap
- Verify the performance of the floor – the time is taken to go to a floor
- Verify that in case of power failure, lift doesn't free-fall and get halted in the particular floor
- Verify lifts working in case button to open the door is pressed before reaching the destination floor
- Verify that in case door is about to close and an object is placed between the doors if the doors sense the object and again open or not
- Verify the time duration for which door remain open by default
- Verify if lift interior is having proper air ventilation
- Verify lighting in the lift
- Verify that at no point lifts door should open while in motion
- Verify that in case of power loss, there should be a backup mechanism to safely get into a floor or a backup power supply
- Verify that in case multiple floor number button is clicked, lift should stop at each floor
- Verify that in case of capacity limit is reached users are prompted with warning alert- audio/visual
- Verify that inside lift user are prompted with current floor and direction information the lift is moving towards- audio/visual prompt

57. Write a Scenario of whatsapp Group (generate group)



- Verify that user can click on 3 dots or not.
- Verify that new pop up is open or not.

- Verify that when click on new group is open or not.
- Verify that admin able to add people or not.
- Verify that admin able to add max 1023 people or not.
- Verify that admin able to add people with link or not.

58. Write a Scenario of Whatsapp payment

- Verify that Rs icon is showing or not.
- Verify that user can click or not.
- Verify that add bank account is open or not.
- Verify that user can click on get started or not.
- Verify that user can click on accept and continue or not.
- Verify that user can select bank or not.

