1. **What is Exploratory testing?**

**Exploratory Testing** is a type of software testing where Test cases are not created in advance but testers check the system on the fly. They may note down ideas about what to test before test execution. The focus of exploratory testing is more on testing as a “thinking” activity.

Exploratory Testing is widely used in Agile models and is All about discovery, investigation, and learning. It emphasizes personal freedom and responsibility of the individual tester.

1. **What is a traceability matrix?**

Traceability matrix is a table type document that is used in the development of software applications to trace requirements. It can be used for both forward (from Requirements to Design or Coding) and backward (from Coding to Requirements) tracing. It is also known as Requirement Traceability Matrix (RTM) or Cross Reference Matrix (CRM).

1. **What is Boundary value testing?**

Boundary value testing is the process of testing between extreme ends or boundaries between partitions of the input values.

Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges.

1. **What is Equivalence partitioning testing?**

Equivalence Partitioning is a type of black box testing technique which can be applied to all levels of software testing like unit, integration, system, etc. In this technique, input data units are divided into equivalent partitions that can be used to derive test cases which reduces time required for testing because of the small number of test cases.

In EP we must identify Valid Equivalence partitions and Invalid Equivalence partitions where applicable.

1. **What is Integration testing?**

**Integration Testing** is a 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.

1. **What determines the level of risk?**

A factor that could result in future negative consequences; usually expressed as impact and likelihood

The likelihood of an adverse event and the impact of the event.

1. **What is Alpha testing?**

**Alpha Testing** is a type of software testing performed to identify bugs before releasing the software product to the real users or public. It is a type of [acceptance testing.](https://www.guru99.com/user-acceptance-testing.html) The main objective of alpha testing is to refine the software product by finding and fixing the bugs that were not discovered through previous tests.

1. **What is beta 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.

Direct feedback from customers is a major advantage of Beta Testing. This testing helps to test products in the customer's environment.

1. **What is component testing?**

Component testing is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components. It’s also referred to as Module Testing when it is viewed from an architecture perspective. Component Testing is also referred to as Unit Testing, Program Testing or Module Testing.

1. **What is functional system testing?**

Functional System Testing a requirement that specifies a function that a system or system component must perform .

Functional system testing is basically defined as a type of testing that verifies that each function of the software application works in conformance with the requirement and specification.

1. **What is Non-Functional Testing?**

Non-Functional Testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application.

Non-Functional Testing is the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability

**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 – toolbar, menu bar, dialog boxes and windows etc**.**

**13. What is Adhoc testing?**

Adhoc Testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

Adhoc testing can be achieved with the testing technique called Error Guessing.

**14. What is load testing?**

**Load Testing** is a type of [Performance Testing](https://www.geeksforgeeks.org/performance-testing-software-testing/) that determines the performance of a system, software product, or software application under real-life based load conditions. Basically, load testing determines the behavior of the application when multiple users use it at the same time. It is the response of the system measured under varying load conditions. The load testing is carried out for normal and extreme load conditions.

**15. What is Stress Testing?**

Stress Testing is a type of software testing that verifies stability & reliability of software application. The goal of Stress testing is measuring software on its robustness and error handling capabilities under extremely heavy load conditions and ensuring that software doesn’t crash under crunch situations. It even tests beyond normal operating points and evaluates how software works under extreme conditions.

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

White Box Testing is a testing technique in which software’s internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

White Box testing Type:

* Statement Coverage
* Decision Coverage
* Branch Coverage
* Condition Coverage
* Multiple Condition Coverage
* Finite State Machine Coverage
* Path Coverage
* Control flow testing
* Data flow testing

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

**Black-box testing:** Testing, either functional or non-functional, without reference to the internal structure of the component or system.

Black box testing techniques:

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

**18. Mention what are the categories of defects?**

The categories of defects:

* Arithmetic Defects
* Logical Defects
* Syntax Defects
* Multi-threading Defects
* Interface Defects
* Performance Defects

**19. Mention what bigbang 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?**

Purpose of exit criteria is to define when we STOP testing either at the: End of all testing – i.e. product Go Live End of phase of testing (e.g. hand over from System Test to UAT.

**21. When should "Regression Testing" be performed?**

**Regression testing** is done to ensure that new code changes do not have side effects on the existing functionalities. It ensures that the old code still works once the latest code changes are done.

We do regression testing whenever the production code is modified

1.When new functionality is added to the application.

2.When there is a Change Requirement.

3. When the defect fixed

4.When there is a performance issue fix

5. When there is an environment change

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

There are 7 key principles:

1. Testing shows presence of Defects

2. Exhaustive Testing is Impossible!

3. Early Testing

4. Defect Clustering

5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy

**1.Testing shows presence of Defects:**

Software testing can ensure that defects are present but it can not prove that software is defect-free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

**2. Exhaustive Testing is Impossible!:**

Exhaustive testing is impossible, meaning the software can never test at every test case. It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc., which is impractical.

**3. Early Testing:**

The defect detected in the early phases of SDLC will be very less expensive. For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

**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.

**5. The Pesticide Paradox:**

Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.

**6. Testing is context-dependent:**

The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.

**7**. **Absence of errors fallacy:**

If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfill all the customer requirements.

**23. Difference between QA v/s QC v/s Testing.**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N** | **Quality Assurance** | **Quality Control** | **Testing** |
| 1 | Focuses on processes and procedures rather than conducting actual testing on the system. | Focuses on actual testing to identify bugs/defects through implementation of procedures and processes. | Focuses on actual testing. |
| 2 | It is process oriented activities. | It is product oriented activities. | Product oriented activities. |
| 3 | It is a preventive activity. | It is a corrective activity. | It is a preventive activity. |
| 4 | 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 testing.**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Smoke Testing** | **Sanity Testing** |
| 1 | Smoke Testing is a software testing technique performed post software build to verify that the critical functionalities of software are working fine. | Sanity Testing is done to check the new functionality/bugs have been fixed. |
| 2 | Smoke Testing has a goal to verify “stability” | Sanity Testing has a goal to verify “rationality”. |
| 3 | Smoke Testing is done by both developers or testers. | Sanity Testing is done by testers. |
| 4 | Smoke Testing verifies the critical functionalities of the system. | Sanity Testing verifies the new functionality like bug fixes. |
| 5 | Smoke testing is a subset of acceptance testing. | Sanity testing is a subset of Regression Testing. |
| 6 | Smoke testing is documented or scripted. | Sanity testing isn’t documented or scripted. |
| 7 | Smoke testing verifies the entire system from end to end. | Sanity Testing verifies only a particular component. |
| 8 | Smoke testing is like General Health Check Up | Sanity Testing is like specialized health check up |

**25. Difference between verification and Validation.**

|  |  |  |
| --- | --- | --- |
| **S.N** | **Verification** | **Validation** |
| 1 | The verifying process includes checking documents, design, code, and program. | It is a dynamic mechanism of testing and validating the actual product. |
| 2 | It does ***not*** involve executing the code | It always involves executing the code |
| 3 | Verification uses methods like reviews, walkthroughs, inspections, and desk- checking etc. | It uses methods like Black Box Testing,White Box Testing, and non-functional testing. |
| 4 | Verification is done by the QA team. | Validation is done by the involvement of the testing team with the QA team. |
| 5 | Verification process comes before validation. | Validation process comes after verification. |

**26. Explain types of Performance testing.**

Performance Testing is a software testing process used for testing the speed, response time, stability, reliability, scalability, and resource usage of a software application under a particular workload.

Types of performance testing:

* **Load testing –** checks the application’s ability to perform under anticipated user loads. The objective is to identify performance bottlenecks before the software application goes live.
* **Stress testing –** involves testing an application under extreme workloads to see how it handles high traffic or data processing. The objective is to identify the breaking point of an application.
* **Endurance testing –** is done to make sure the software can handle the expected load over a long period of time.
* **Spike testing –** tests the software’s reaction to sudden large spikes in the load generated by users.
* **Volume testing** – Under Volume Testing large no. of. Data is populated in a database, and the overall software system’s behavior is monitored. The objective is to check software application’s performance under varying database volumes.
* **Scalability testing** – The objective of scalability testing is to determine the software application’s effectiveness in “scaling up” to support an increase in user load. It helps plan capacity additions to your software system.

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

**Error:-** A mistake made by humans during coding is called an error

**Defect:-** A problem in the functioning of a software system during testing that is called defect

**Bug:-** a defect to be accepted by the development team is called a bug

**Failure:-** when a build does not meet its specifications then it is termed as failure.

**28. Difference between Priority and Severity.**

|  |  |
| --- | --- |
| **Priority** | **Severity** |
| Priority is the order in which the developer should resolve a defect | Severity is the degree of impact that a defect has on the operation of the product. |
| Priority is categorized into three types: low, medium and high | Severity is categorized into five types: critical, major, moderate, minor and cosmetic. |
| 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 status is based on customer requirements | Severity status is based on the technical aspect of the product |
| High Priority and low severity status indicates, defect have to be fixed on immediate basis but does not affect the application | High Severity and low priority status indicates defects have to be fixed but not on immediate basis. |

**29. What is the Bug Life Cycle?**

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Bug Life Cycle’

The purpose of Defect/Bug life cycle is to easily coordinate and communicate current status of defect which changes to various assignee and make the defect fixing process systematic and efficient.

**30.Explain the difference between Functional testing and Non Functional testing.**

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| It verifies the operations and actions of an application. | It verifies the behavior of an application. |
| It is based on the requirements of the customer. | It is based on the expectations of customers. |
| It helps to enhance the behavior of the application. | It helps to improve the performance of the application |
| It tests what the product does. | It describes how the product does. |
| Functional testing is based on the business requirement. | Non-functional testing is based on the performance requirement |
| Examples:  1. Unit Testing  2. Smoke Testing  3. Integration Testing  4. Regression Testing | Examples:  1. Performance Testing  2. Load Testing  3. Stress Testing  4. Scalability Testing |

**31. To create HLR & TestCase of Instagram and Facebook only first page.**

|  |  |
| --- | --- |
| **HLR-Instagram** | [**Assignment\_M2\_HLR**](https://docs.google.com/spreadsheets/u/0/d/1i-kROXWRKiOdvS-lOcpigU__BIVNLAWHCleySlJAel0/edit) |
| **HLR-Facebook** | [**Assignment\_M2\_HLR**](https://docs.google.com/spreadsheets/u/0/d/1i-kROXWRKiOdvS-lOcpigU__BIVNLAWHCleySlJAel0/edit) |

|  |  |
| --- | --- |
| **Test case-Instagram** | [**Assignment\_M2\_TESTCASE**](https://docs.google.com/spreadsheets/u/0/d/12tNv1Y4im5cvG5YJsOXNV8SJquomcTWgaE4k29-KASo/edit) |
| **Test case-Facebook** | [**Assignment\_M2\_TESTCASE**](https://docs.google.com/spreadsheets/u/0/d/12tNv1Y4im5cvG5YJsOXNV8SJquomcTWgaE4k29-KASo/edit) |

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

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| SDLC is mainly related to software development. | STLC is mainly related to software testing. |
| Besides development other phases like testing is also included. | It focuses only on testing the software. |
| SDLC involves a total of six phases or steps. | STLC involves only five phases or steps. |
| In SDLC, more members (developers) are required for the whole process. | In STLC, less number of members (testers) are needed. |
| Goal of SDLC is to complete successful development of software. | Goal of STLC is to complete successful testing of software. |
| It helps in developing good quality software. | It helps in making the software defects free. |

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

**Test scenario:-**

A Test Scenario is defined as any functionality that can be tested. It is also called Test Condition or Test Possibility.

**Test cases:-**

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, post condition developed for specific test scenarios to verify any requirement. The test case includes specific variables or conditions.

**Test scripts:-**

Test Scripts are a line-by-line description containing the information about the system transactions that should be performed to validate the application or system under test. Test script should list out each step that should be taken with the expected results.

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

Test Plan is A document describing the scope, approach, resources, and schedule of intended test activities.

The information should be covered including:

1. Test Planning
2. Test Plan strategy
3. Test Planning Factors
4. Test Planning Activity
5. Exit criteria

**35. What is priority?**

Priority is important for fixing the bug or which bug to be fixed first or how soon the bug should be fixed.

**36. What is severity?**

The impact of the bug on the application is known as severity.

**37. Bug categories are…**

**Data Quality/Database Defects:** Deals with improper handling of data in the database**.**

**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.

**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.

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

**38. Advantage of Bugzilla .**

➝ It improves the quality of the product.

➝ It enhances the communication between the developing team and the testing team.

➝ It has the capability to adapt to multiple situations

**39**. **What are the different Methodologies in Agile Development Model?**

It is an Iterative and Incremental Approach.

Agile is an Iterative and Incremental Process

The different Methodologies in Agile Development Model is:

1. Scrum

### Kanban

### Extreme Programming (XP)

**1. Scrum:-**

SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team based development environment. Basically, Scrum is derived from activity that occurs during rugby matches. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members).

It consists of three roles and their responsibilities are explained as follows:

**Scrum Master:** Master is responsible for setting up the team, sprint meeting and removes obstacles to progress

**Product owner:** The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration

**Scrum Team:** Team manages its own work and organizes the work to complete the sprint or cycle

Scrum also features a robust set of principles and activities that dictate how you work. These include:

**Sprint planning:** Planning sessions to identify the purpose behind your sprints

**Roles:** Key roles in the Scrum project management process

**Product backlog:** A list of tasks arranged according to priority level

**2. Kanban:-** Kanban is a very popular framework for development in the agile software development methodology. It provides a transparent way of visualizing the tasks and work capacity of a team. Kanban can be used in any domain, and it can be used very effectively in software development. Kanban should be used when you want to release your work at any time. It requires

git branching, but it is doable.

**3. Extreme Programming:-**

Extreme Programming (XP) was designed for Agile software development projects. It focuses on continuous development and customer delivery and uses intervals or sprints, similar to a Scrum methodology.Extreme Programming puts the customer at the center of everything the team does. In XP, teams can involve the end users in a meaningful way and use the feedback they get to deliver the best possible product.

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

**Authentication:-** Authentication is the process of identifying someone's identity by assuring that the person is the same as what he is claiming for.

In this, the user or client and server are verified.

**Authorization** :- Authorization is the process of giving permission to access the resources.

In this, it is verified that the user is allowed through the defined policies and rules.

One of the main challenges of web application testing is ensuring that your web app works well across different browsers, devices, and operating systems. Different browsers may have different rendering engines, standards support,

features, and extensions that can affect how your web app looks and behaves.

**41. To create HLR & Test Case of Web Based (Whats App web , Instagram)**

|  |  |
| --- | --- |
| **HLR Web Based Whats app web** | [**Assignment\_M2\_HLR**](https://docs.google.com/spreadsheets/u/0/d/1i-kROXWRKiOdvS-lOcpigU__BIVNLAWHCleySlJAel0/edit) |
| **HLR Web Based Instagram** | [**Assignment\_M2\_HLR**](https://docs.google.com/spreadsheets/u/0/d/1i-kROXWRKiOdvS-lOcpigU__BIVNLAWHCleySlJAel0/edit) |

|  |  |
| --- | --- |
| **Test case Web Based Whats app web** | [**Assignment\_M2\_TESTCASE**](https://docs.google.com/spreadsheets/u/0/d/12tNv1Y4im5cvG5YJsOXNV8SJquomcTWgaE4k29-KASo/edit) |
| **Test case Web Based Instagram** | [**Assignment\_M2\_TESTCASE**](https://docs.google.com/spreadsheets/u/0/d/12tNv1Y4im5cvG5YJsOXNV8SJquomcTWgaE4k29-KASo/edit) |

**42. To create HLR and Test Case on this Link.** [**https://artoftesting.com/**](https://artoftesting.com/)

|  |  |
| --- | --- |
| **HLR - Artoftesting** | [**Assignment\_M2\_HLR**](https://docs.google.com/spreadsheets/u/0/d/1i-kROXWRKiOdvS-lOcpigU__BIVNLAWHCleySlJAel0/edit) |
| **Test Case - Artoftesting** | [**Assignment\_M2\_TESTCASE**](https://docs.google.com/spreadsheets/u/0/d/12tNv1Y4im5cvG5YJsOXNV8SJquomcTWgaE4k29-KASo/edit) |

**43. Write a scenario of only Whats app chat messages**

**Whats app chat messages scenario**

1. Verify that the user can send messages to any individual selected from his contact list.

2. Verify that the ‘Chats’ window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.

3. Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person.

4. Verify that the user can check the message delivered and read the time for a message in the ‘Message Info’ section.

5. Verify that the user can share or receive contact with the other person.

6. Verify that the user can create a group by adding multiple people from his contact list.

7. Verify that the user can send and receive the message in group chats.

8. Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals.

9. Verify that users can send and receive images, audio, video, and emoticons in group chats.

10. Verify that the user can send and receive chats in the secondary languages available.

11. Verify that users can delete text, images, audio, and video messages within a chat.

12. Verify that users can clear their complete chat history in an individual or group chat.

13. Verify that users can archive chats in an individual or group chat.

14. Verify that the user can check data usage by images, audio, video, and documents in Whats App chats.

15. Verify that the user can set a chat wallpaper.

16. Verify that the user can show previews for both group and individual chats.

**44. Write a Scenario of Pen**

**Pen Scenario**

1.Verify if you are able to hold the pen comfortably.

2. Verify if you are able to write smoothly.

3. Verify that the pen is not making any sound while writing.

4. Verify the ink flow. It should not overflow nor get a break either.

5. Verify the quality of the material used for the pen.

6. Verify if the company or pen name is visible clearly.

7. Verify if the pen color or text written on the pen is not getting removed easily.

8. Verify, whether the width of the line drawn by the pen is as per the expectations or not.

9. Verify the ink color, it should be consistent from the start till the end.

10. Verify if a pen can write on a variety of papers like smooth, rough, thick, thin, glossy etc.

11. Verify for the waterproof ink.

12. Verify if the ink will not get dried easily by keeping the pen open for some time.

13. Verify if any other refill fits in the pen or not.

14. Verify that the pen doesn’t have sharp edges or corners.

15. Verify if the ink and external assembly of the pen is made of non-toxic material.

**45. Write a Scenario of Pen Stand**

**Pen stand**

1. Verify the quality of the material used for the pen stand.

2. Verify that pen stands are able to hold the pen comfortably.

3. Verify if the company logo or pen stand name is visible clearly.

4. Verify that the pen stand doesn’t have sharp edges or corners.

5. Verify that the pen stand material is durable or not.

6. Verify that the pen stand color, size, height, width is as per specification or not.

7. Verify that the pen stand is made of ceramic, glass, cardboard, wood, metal or enameled metal.

8. Verify that the pen stand usually has a mouth of the same width as the base to keep as many pens at a time.

9. Verify that the pen stand is waterproof or not.

10. Verify that the pen stand is flat or not.

11. Verify that the pen stand is spherical, hexagonal, rectangular, pentagonal or octagonal in shape.

**46. Write a Scenario of Door**

**Door Scenario**

1.Verify if the door is a single door or bi-folded door.  
2. Check if the door opens inwards or outwards.  
3. Verify that the dimension of the doors are as per the specifications.  
4. Verify that the material used in the door body and its parts is as per the specifications.  
5. Verify that the color of the door is as specified.  
6. Verify if the door is a sliding door or rotating door.  
7. Check the position, quality and strength of hinges.  
8. Check the type of locks in the door.  
9. Check the number of locks in the door interior side or exterior side.  
10. Verify if the door is having a peek-hole or not.

11. Verify if the door has a stopper or not.  
12. Verify if the door closes automatically or not – spring mechanism.  
13. Verify if the door makes noise when opened or closed.  
14. Check the door condition when used extensively with water.  
15. Check the door condition in different climatic conditions- temperature, humidity etc.  
16. Check the amount of force- pull or push required to open or close the door.

**47. Write a Scenario of ATM**

**ATM Scenario**

1.Verify that all buttons, images, labels and text boxes are displayed on the screen.

2. Verify that the informative text displayed on the screen is clearly visible or not.

3. Verify that color, size and other all objectives properly as per the specification.

4. Verify that the type of ATM machine,if it is a touch screen, keyboard buttons only, both.

5. Verify that on properly inserting valid cards different banking options appear or not.

6. Verify that the touch of the ATM screen is smooth and operational.

7. Check the card is inserted incorrectly in the machine, no option to continue and enter credential is displayed to the user.

8. Verify that the user is presented with the option to choose the language for the further operation.

9. Check the user is asked the pin number before displaying any bank/card details.

10. Verify that there is a limited number of attempts to user enter the pin code.

11. Check that the user entered the pin number when the pin is masked.

12. Verify that the user is presented with a different type of account option displayed like- saving and current.

13. Check the user entered the incorrect pin , the displayed error message on the screen.

14. Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.

15. Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.

16. Check that the user cannot withdraw more than amount the total available balance and a proper message should be displayed.

17. Verify that the user is provided the option to get the transaction details in printed forms.

18. Check that in case the ATM machine runs out of money, a proper message is displayed to the user.

19. Check the user is not allowed to proceed with the expired ATM card and that proper error message gets displayed.

20. 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.

**48. When to use Usability Testing?**

The goal of usability testing is to understand how real users interact with your website and make changes based on the results.

**49. 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 the 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 a 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.

**50. Write a scenario of Microwave Oven**

**Microwave Oven Scenario**

1.Check the oven body provided as per the specification.

2. Check the oven functionality with maximum temperature.

3. Check the oven functionality with minimum temperature.

4. Check the oven to properly heat the food at the desired temperature.

5. Check the oven and heat the food at the desired temperature duration.

6. Check the oven plate rotation speed is optimal or not too high.

7. Check the oven’s door is closed properly.

8. Check the oven’s door opens smoothly.

9. Check the oven’s power supply properly.

10. Check the oven’s functionality with different kinds of food.

11. Check the oven’s functionality with different kinds of food at different temperatures.

12. Check the oven’s functionality with different container material.

13. Check the oven’s digital display clearly or not.

14. Verify that the temperature regulator works correctly.

15. Verify that the temperature regulator is smoothly movable or not.

16. Check if the oven’s door handle is durable or not.

17. Verify that the power cord of the oven is long enough.

**51. Write a scenario of Coffee vending Machine**

**Coffee vending Machine Scenario**

1.Verify that the machine body is as per the specification.

2. Verify that the machine inner part is as per the specification.

3. Verify the machine power/voltage requirements.

4. Verify that you press all buttons easily.

5. Verify that the presses all buttons are working or not.

6. Verify that the display is displayed correctly.

7. Verify that the quantity of hot water, milk,coffee powder per serving is correct or not.

8. Verify that the quantity of coffee ingredients runs out to display errors on screen.

9. Verify that the machine company logo and color are visible as per the specification.

10. Verify that the machine is switched off so the coffee does not come out of the nozzle.

11. Verify that the machine nozzle is not leaking.

12. Verify that the machine is switched off and the light is turned on properly.

13. Verify that the machine is switched on and the light is turned on properly.

14. Verify that the machine should get the correct temperature and same temperature each time coffee is served.

15. Verify that the button is pressed multiple times to lead to multiple serving of coffee.

16. Verify that the machine should work correctly in different climates.

17. Verify that the machine should not make much noise.

**52. Write a scenario of chair**

**Chair Scenario**

1. Verify that the chair is stable enough to take an average human load.

2. Check the material used in making the chair-wood, plastic etc.

3. Check if the chair’s legs are level to the floor.

4. Check the usability of the chair as an office chair, normal household chair.

5. Check if there is back support in the chair.

6. Check if there is support for hands in the chair.

7. Verify the paint’s type and color.

8. Verify if the chair’s material is durable or not.

9. Check if the cushion is provided with a chair or not.

10. Check the condition when washed with water or the effect of water on the chair.

11. Verify that the dimension of the chair is as per the specifications.

12. Verify that the weight of the chair is as per the specifications.

13. Check the height of the chair’s seat from the floor.

**53. To Create Scenario (Positive & Negative)**

**1. Facebook Chat on Mobile**

**2. G mail(Receiving mail)**

**3. Online shopping to buy product (flip kart)**

1. **Facebook Chat on Mobile**

1. Verify that the user can send messages to any individual selected from his contact list.

2. Verify that the ‘Chats’ window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated.

3. Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person.

4. Verify that the user can check the message delivered and read the time for a message in the ‘Message Info’ section.

5. Verify that the user can share or receive contact with the other person.

6. Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals.

7. Verify that the user can send and receive chats in the secondary languages available.

8. Verify that users can delete text, images, audio, and video messages within a chat.

9. Verify that users can clear their complete chat history in an individual chat.

10. Verify that users can archive chats in an individual chat.

11. Verify that the user can set a chat wallpaper.

12. Verify that the user can show previews for both individual chats.

1. **G mail(Receiving mail)**

1. Verify that a newly received email is displayed as highlighted in the Inbox section.

2. Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line).

3. Verify that on clicking the newly received email, the user is navigated to email content.

4. Verify that the email contents are correctly displayed with the desired source formatting.

5. Verify that any attachments are attached to the email and are downloadable.

6. Verify that the attachments are scanned for viruses before download.

7. Verify that all the emails marked as read are not highlighted.

8. 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.

9. Verify that count of unread emails is displayed alongside ‘Inbox’ text in the left sidebar of G mail.

10. Verify that unread email count increases by one on receiving a new email.

11. Verify that unread email count decreases by one on reading an email ( marking an email as read).

12. Verify that email recipients in cc are visible to all users.

13. Verify that email recipients in bcc are not visible to the user.

14. Verify that all received emails get piled up in the ‘Inbox’ section and get deleted in cyclic fashion based on the size availability.

15. Verify that email can be received from non-g mail email Ids like – yahoo, Hot mail etc

1. **Online shopping to buy product (flip kart)**

1. Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.

2. Verify that the user can add to the cart one or more products.

3. Verify that users can add products to the wish list.

4. Verify that the user can see the previously added products on the cart page, after signing in to the application.

5. Verify that the user can successfully buy more than one product that was added to his/her cart.

6. Verify that the user cannot add more than the available inventory of the product.

7. Verify that the limit to the number of products a user can buy is working correctly. Also, an error message gets displayed, preventing the user from buying more than the limit.

8. Verify that the delivery can be declined during checkout for the places where shipping is not available.

9. Verify that the Cash on Delivery option of payment is working fine.

10. Verify that the different prepaid methods of payments are working fine.

11. Verify that product return functionality works correctly.

**54. Write a Scenario of Wrist Watch**

**Wrist Watch Scenario**

1.Verify the type of watch – analog or digital.

2. In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.

3. In the case of a digital watch, check if the digital display for hours, minutes, and seconds is correctly displayed.

4. Verify the material of the watch and its strap.

5. Check if the shape of the dial is as per specification.

6. Verify the dimension of the watch is as per the specification.

7. Verify the weight of the watch.

8. Check if the watch is waterproof or not.

9. Verify that the numbers in the dial are clearly visible or not.

10. Check if the watch has a date and day display or not.

11. Verify the color of the text displayed in the watch – time, day, date, and other information.

12. Verify that the clock's time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.

13. Check if the second hand of the watch makes a ticking sound or not.

14. Verify the brand of the watch and check if it's visible in the dial.

15. Check if the clock is having stopwatch, timers, and alarm functionality or not.

16. In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.

17. Verify if the watch comes with any guarantee or warranty.

18. Verify if the dial has glass covering or plastic, check if the material is breakable or not.

19. Verify if the dial’s glass/plastic is resistant to minor scratches or not.

20. Check the battery requirement of the watch.

**55. Write a Scenario of Lift(Elevator)**

**Lift Scenario**

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

**56. Write a Scenario of whats app Group (generate group)**

**whats app Group (generate group) scenario**

1. Check that the user can create a Whats app group.

2. The user can set a name for the created group.

3. Check that the user can add and save the group description.

4. Verify that the user can make multiple people as group Admin.

5. Check that only group admins can add people to the group.

6. Is there any option to mute group notifications for some time?

7. Is there any option to add people to the group by sharing a link?

8. The admin can delete users from this group.

9. Check that admin can change settings like only admin can share information in this group, or everyone can share information in this group.

10. If a person is removed from the group, he/she will not be able to see any updates.

11. Check that the user can exit the group by clicking on the exit button or not.

12. Verify that the admin can delete the group.

13. Check that you are able to create a group without adding any members.

**57. Write a Scenario of Instagram ( video call with chat )**

**Instagram(video call with chat)**

1. Verify that the user can video call with how many people join in video call one at a time.

2. Verify that the user should click on the video call button and start video call or not.

3. Verify that the user is calling the video and call them if an individual person or group will receive a notification or not.

4. Verify that the user can accept the video call or not.

5. Verify that the user can click on full screen or small screen in the video call screen or not.

6. Verify that the user can set the camera setting or not.

7. Verify that the user can end the video call or not.

8. Verify that the user can receive the video call when clicking on the join button or not.

9. Verify that the user can join a video call if the user should not be in an Instagram account .

10. Verify that users can watch posts and reels in a video call with others from the Instagram app for Android.

11. Verify that the user can chat with a video call or not.

12. Verify that the user can stop viewing content together in your video call or not.

13. Verify that the user can sharing your screen in a video call or not.

14. Verify that the user can stop sharing your screen in a video call or not.

15. Verify that the user can check whether your camera, microphone and speaker are working or not.

16. Verify that the user can check if all group members do chat or not.

17. Verify that the user can send the message,video,audio and emojis with video call or not.

**58. Write a Scenario of Whats app payment**

**Whats app payment**

1. Verify that the user already has an UPI ID and bank accounts.

2. Verify that the opposite user is using whats app payment.

3. Verify that the user can select a user and make a payment or/not.

4. Verify that the user can pay money securely or/not.

5. Verify that the user can see the opposite user’s username.

6. Verify that the user can see his/her bank account and other payment methods.

7. Verify that the user can add other payment methods or/not.

8. Verify that the user can see his/her available account balance.

9. Verify that the user can request money from the opposite user.

10. Verify that the user can write a number of amounts.

11. Verify that the user can write in the description with payment.

12. Verify that the user sends a payment and the opposite user has received it or/not.