**Que.1What is Exploratorytesting ?**

* We have to explore the application to understand the requirement completely, then only we will start testing. In simple words, To understand the requirement completely, first we have to explore the application, hence it is called as exploratory. Test engineer will do testing, when there is no requirement at all or the requirements are missing.
* the current trend in testing is to push for automation, exploratory testing is a new way of thinking. Automation has its limits. Is not random testing but it is Adhoc testing with purpose of find bugs Is structured and rigorous Is cognitively (thinking) structured as compared to procedural structure of scripted testing

**Que.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)
* Simple technique may be quite valuable (naming convention)
* Forward Traceability – Mapping of Requirements to Test cases
* Backward Traceability – Mapping of Test Cases to Requirements
* Bi-Directional Traceability - A Good Traceability matrix is the References from test.

**Que.3What 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

**Que.4What 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.
* The numbers fall into a partition where each would have the same, or equivalent, result i.e. an Equivalence Partition (EP) or Equivalence Class
* The Valid partition is bounded by the values 1 and 100
* Plus there are 2 Invalid partitions

**Que.5what is integration testin ?**

* 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.
* Components may be code modules, operating systems, hardware and even complete systems
* There are 2 levels of Integration Testing.

1. Component Integration Testing

2. System Integration Testing

**Que.6 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
* 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.
* It is conducted for the software application and project.

**Que.7What is beta testing?**

* Beta Testing is always open to the market and public
* It is performed in Real Time Environment
* It is always performed by the customers at their own site.
* 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 also the form of Acceptance Testing
* It is only a kind of Black Box Testing.

**Que.8 what is component testing?**

* 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
* Sometimes known as Unit Testing, Module Testing or Program 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.
* Unit testing in Extreme Programming involves the extensive use of testing frameworks. A unit test framework is used in order to create automated unit tests. Unit testing frameworks are not unique to extreme programming, but they are essential to it.
* Component testing is like unit testing with the difference that the developer uses real deta instead ofdummy7 deta for testing of the written code.
* Unit testing is performed by using the White Box Testing method.

**Que.9 what is functional system testing ?**

* Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system.
* Functional testing verifies that each function of the software application operates in conformance with the requirement specification This testing mainly involves black box testing and it is not concerned about the source code of the application.
* Manual testing or automation tools can be used for functional testing
* Business requirements are the inputs to functional testing
* Easy to do manual testing
* Functional testing is executed first
* Types of Functional testing are
* Unit Testing
* Smoke Testing
* Sanity Testing
* Integration Tasting
* White box testing
* ∙ Black Box testing
* User Acceptance testing
* Regression Testing
* Examples
* *{Desktop Based Testing :}*

Check for broken links. Warning messages. Resolution change effect on the application

**Que.10 what is non-functional testing?**

* Non-Functional Testing: Testing the attributes of a component or system that do not relate to functionality,
* reliability, efficiency, usability, interoperability, maintainability and portability
* To address this issue, performance testing is carried out to check & fine tune system response times
* It is the testing of “how” the system works.
* Non-functional testing describes how good the product works
* Non- functional testing should be performed after functional testing
* Tough to do manual testing
* **Examples**
* *{Mobile Based Testing : }*

In mobile ,automatically will switch off without any reason. To stop the application which is not in our hand.

Que.11 what is GUL 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
* Examples
* *{Mobile Based Testing}* :

If mobile is in every orientation mode so display image, video properly.

Every app will display in responsive type.

Alignment should be apply properly of every field.

Que.12 What isadhoc testing?

* The Error guessing is a technique where the 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.
* Use Error Guessing to Complement Test Design Techniques
* Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.
* Adhoc testing can be achieved with the testing technique called Error Guessing
* There are different types of Adhoc testing and they are listed as below:
* Buddy Testing
* Pair testing
* Monkey Testing

Que. 13 what load testing?

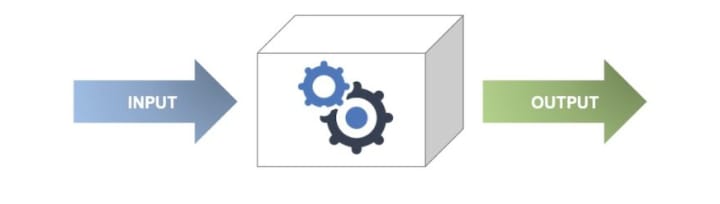
* Load testing – It’s a performance testing to check system behavior under load.
* Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions.
* Load testing identifies the bottlenecks in Stress testing determines the the system under various workloads checks how the system reacts when the load is gradually increased
* Load testing does not break the system
* Examples
* An Airline website was not able to handle 10000+ users during a festival offer.
* Encyclopedia Britannica declared free access to their online database as a promotional offer. They were not able to keep up with the onslaught of traffic for weeks.

**Que.14 what is 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.
* It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions.
* Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing.
* Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks.
* Examples
* During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.

**Que.15 what is white box testing and list the type of white box testing?**

* 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.
* White box testing is also called glass testing or open box testing.
* The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.



* *Examples*
* Mobile Based Testing :

The Android SDK and related plugin for Eclipse

Android devices enabled for development and debugging, with appropriate USB drivers as necessary

* Game Based Testing :

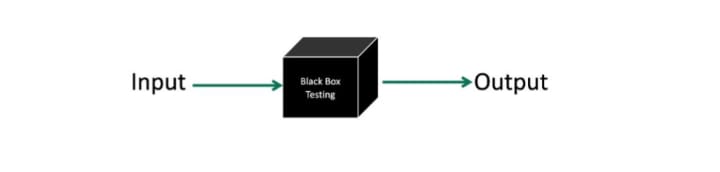
When we connect with remote device,so which device we connect will check in code When some debug the code and play at that time game.

* Desktop Based Testing

When we debug the code when we writing

**Que.16 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
* The testers have no knowledge of how the system or component is structured inside the box.
* The tester is oblivious to the system architecture and does not have access to the source code
* Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.



* Examples
* Desktop Based Testing

Resolution change effect on the application

Installation Testing (Upgrade/Downgrade)

* Mobile Based Testing

In mobile, automatically will switch off without any reason.

To stop the application which is not in our hand.

**Que.18 mention what bigbang testing is?**

* In Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
* Big Bang testing has the advantage that everything is finished before integration testing starts. The major disadvantage is that in general it is time consuming and difficult to trace the cause of failures because of this late integration.
* Here all component are integrated together at once, and then tested.
* Advantages
* Convenient for small systems
* Disadvantages
* Fault Localization is difficult.
* Given the sheer number of interfaces that need to be tested in this approach, some interfaces links to be tested could be missed easily.
* Since the integration testing can commence only after “all” the modules are designed, testing team will have less time for execution in the testing phase

**Que.20 when should “regression testing “be performed?**

* Re-testing: Testing that runs test cases that failed the last time they were run, in order to verify the success of corrective actions
* Whenever a fault is detected and fixed then the software should be re-tested to show that the original fault has been fixed. This is known as Re-Testing. It is important that the test case is repeatable.
* If the test is re-run and passes you cannot necessarily say the fault has been resolved because .You also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements – Regression Testing
* when the system is stable and the system or the environment changes when testing bug-fix releases as part of the maintenance phase
* It should be considered complete when agreed completion criteria for regression testing have been met Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation

**Que.22 what is 7key principles? Explain in detail?**

* 7key principles
* Testing shows presence of Defects
* Exhaustive Testing is Impossible
* Early Testing
* Defect Clustering
* The Pesticide Paradox
* Testing is Context Dependent
* Absence of Errors Fallacy
* ***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
* ***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.
* 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
* hat is we must Prioritise our testing effort using a Risk Based Approach
* **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
* **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’
* Similarly, most operational failures of a system are usually confined to a small number of modules An important consideration in test prioritisation!
* **The Pesticide Paradox**
* 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
* 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 improve
* s As software improves the effectiveness of previous tests erodes
* **Testing is Context Dependent**
* Testing is basically context dependent. Testing is done differently in different contexts
* Different kinds of sites are tested differently
* For example
* Safety – critical software is tested differently from an e-commerce

Site

* 1 to 3 failures per KLOC typical for industrial software
* 0.01 failures per KLOC for NASA Shuttle code
* Also different industries impose different testing standards
* Absence of Errors Fallacy
* If the system built is unusable and does not fulfil 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 fulfil the users’ needs and expectations

**Que.23 difference between QA v/s QC V/S testing?**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N** | **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/error/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 bug/defect through implementation of procedures and process. | Focuses on actual testing. |
| **3** | Process oriented activities | Process oriented activities | Process oriented activities |
| **4** | Preventive activities. | It is a corrective process | It is a preventive process |
| **5** | It is a subset of Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance | Testing is the subset of Quality Control. |

**Que.24 Difference between smoke and sanity?**

|  |  |  |
| --- | --- | --- |
| **s.n** | **Smoke testing** | **Sanity testing** |
| **1** | Smoke Testing is performed after software build to ascertain that the critical functionalities of the program is working fine. | After receiving a software build, with minor changes in code, or functionality, Sanity testing is performed to ascertain that the bugs have been fixed and no further issues are introduced due to these changes. |
| **2** | Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |
| **3** | Smoke testing is usually documentedor scripted | Sanity testing is usually not documented andis unscripted |
| **4** | Smoke testing is like General Health Check Up | Sanity Testing is like specialized healthcheck up |

**Que.25 Explain types of performance testing?**

* Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload.
* The focus of Performance testing is checking a software programs
* **Speed –** Determines whether the application responds quickly
* **Scalability –** Determines maximum user load the software application can handle.
* **Stability –** Determines if the application is stable under varying loads

Types of Performance Testing

1. Load testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing

**Que. 26 what is error, defect, bug, and failure?**

* **Error :**A mistake in coding is called error
* **Defect:**error found by tester is called defect,
* **Bug:**defect accepted by development team then it is called bug
* Failure:build does not meet the requirements then it is failure

**Que.27 Difference between priority and severity?**

|  |  |  |
| --- | --- | --- |
| **S.N** | **PRIORITY** | **SECERITY** |
| **1** | Severity is a term that denotes how severely a defect can affect the functionality of the software. | Priority is a term that defines how fast we need to fix a defect. |
| **2** | The value of severity is objective. | The value of priority is subjective. |
| **3** | The value of Severity changes continually from time to time. | The value of Priority changes from time to time |

**Que.28 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.”
* 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 ‘Defect Life Cycle’.
* The process by which the defect moves through the life cycle is depicted next slide.

**Que. 29 explain the difference between function and non-functional testing?**

|  |  |  |
| --- | --- | --- |
| s.n | **functional** | **Non-functional** |
| **1** | Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system. | 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 |
| **2** | Functional testing is executed first | Non functional testing should be performed after functional testin |
| **3** | Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| **4** | Functional testing describes what the product does | Non-functional testing describes how good the product works |

**Que.29 to create HLR and Test case.**

* **Instagram**

<https://in.docworkspace.com/d/sIOf4neHQAefvrqUG?sa=share.copy_link>

* **Facebook**

<https://in.docworkspace.com/d/sIGH4neHQAcDprqUG?sa=share.copy_link>

**Que.30 what is difference between the STLC and SDLC?**

|  |  |  |
| --- | --- | --- |
| **S.N** | **STLC** | <https://web.whatsapp.com/6cb66aae-3460-4335-976b-f14c9405dca4>**SDLC** |
| **1** | . STLC is mainly related to software testing. | SDLC is mainly related to software development |
| **2** | STLC involves only five phases or steps | SDLC involves total six phases or steps. |
| **3** | It helps in developing good quality software. | t helps in making the software defects free. |
| **4** | Creation of reusable software systems is the end result of SDLC. | A tested software system is the end result of STLC. |

**Que. 31 what the difference between tesr case,test scenarios,test script ?**

|  |  |  |  |
| --- | --- | --- | --- |
| **s.n** | **Test Scenario** | **Test Case** | **Test script** |
| **1** | A test scenario contains high-level documentation which describes an end to end functionality to be tested. | Test cases contain definite test steps, data, expected results for testing all the features of an application | A test script in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected |
| **2** | It focuses on more “what to test” **than** “how to test”. | A complete emphasis on “what to test” **and** “how to test.”. | One script is written to explain how to simulate each business scenario |
| **3** | Test scenarios are derived from test artifacts like BRS, SRS, etc. | Test case is mostly derived from test scenarios. Multiple Test case can be derived from a single Test Scenario | The Test Script can be manual or automated |
| **4** | It helps in an agile way of testing the end to end functionality | It helps in exhaustive testing of an application | The Test Procedures Specification specifies the sequence of actions for a test, i.e. one or more Test Cases. It is also known as a Test Script |

**Que. 32 explain what tast pan is ? what is the infromatin thet should be covered?**

* A test plan is a document that outlines the overall approach, objectives, scope, and schedule for testing a software application or system. It serves as a roadmap for the testing process and provides guidance to the testing team on how to proceed with their activities. The primary purpose of a test plan is to ensure that the software or system is thoroughly tested to meet the desired quality standards

=> Here are some key pieces of information that should be covered in a test plan:

* Introduction: This section provides an overview of the software or system being tested, including its purpose, objectives, and any relevant background information
* Test Objectives: It outlines the specific goals and objectives of the testing effort. This helps to align the testing activities with the overall project goals and ensure that the testing is focused and targeted.
* Test Strategy: The test strategy outlines the overall approach and methodologies to be employed during testing. It includes details on the types of tests to be conducted, such as functional testing, performance testing, security testing, etc., and the testing techniques to be used.
* Test Execution: This section covers the actual execution of tests, including the responsibilities of the testing team, the test schedule, and the criteria for determining when testing is complete

**Que33. 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.
* 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

**Que.35 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.

**Que.36bug cetagorice are...?**

* ritical: Bugs that cause the software to crash or malfunction, and need to be fixed urgently.
* Major: Bugs that affect the core functionality of the software, but do not cause it to stop working completely.
* Minor: Bugs that affect the usability or performance of the software, but do not affect the main functionality.
* Trivial: Bugs that are cosmetic or aesthetic, such as typos or alignment issues.
* Functional: Bugs that are related to the functionality of the software, such as buttons not working or applications not running.

**Que.37 Advantage of Bugzila.**

* Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Centre, Test link etc.
* This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data management chain
* Key features of Bugzilla includes
* Advanced search capabilities
* E-mail Notifications

Que.38 **Difference between priority and severity**

|  |  |
| --- | --- |
|  |  |
| Priority is Relative and Business-Focused | Severity is absolute and Customer-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 | 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 |
| 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. | If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by a user is rare but the impact of application crashing is severe. So the severity is high but priority is low |
|  |  |

**Q40. 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

The different Methodologies in Agile Development Model

* Scrum
* Kanban
* DSDM (Dynamic Software Development Method)
* FDD (Feature Driven Development)

**Q41. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

* Authorization and authentication are two essential concepts in web testing that relate to user access and security. Here's an explanation of each term and the differences between them:

1. Authentication: Authentication is the process of verifying the identity of a user or system before granting access to specific resources or functionalities. It ensures that the user is who they claim to be. Common authentication mechanisms include usernames and passwords, biometric authentication (such as fingerprint or facial recognition), tokens, certificates, and single sign-on (SSO) solutions. In web testing, the focus is on verifying that the authentication process works correctly and securely, preventing unauthorized access
2. Authorization: Authorization refers to the process of granting or denying access to specific resources or functionalities based on the authenticated user's permissions and privileges. It determines what actions a user can perform and what data they can access. Authorization is usually based on roles, permissions, and access control lists (ACLs). In web testing, the goal is to ensure that the authorization rules are correctly implemented, and users can only access the resources they are authorized to access

* Common problems faced in web testing include

1. . Performance Issues: Web applications need to handle a large number of concurrent users efficiently. Testing for performance issues such as slow response times, bottlenecks, and scalability problems is crucial to ensure optimal user experience
2. . Broken Links: Links within a web application can become broken or dead over time, leading to a poor user experience. Web testing involves checking for broken links and ensuring that all links navigate correctly.
3. Usability Testing: Usability testing focuses on evaluating the user-friendliness of a web application. It involves assessing the application's intuitiveness, ease of navigation, clarity of instructions, and overall user satisfaction.
4. . Data Validation: Input validation is critical to prevent security vulnerabilities and data integrity issues. Web testing involves checking for proper validation of user input to prevent malicious input or unintended errors

**Q42 To create HLR & TestCase of WebBased (WhatsApp web , Instagram**

* **1. WhatsApp Web :** [**https://web.whatsapp.com/**](https://web.whatsapp.com/)

<https://in.docworkspace.com/d/sIH_4neHQAZT7rqUG?sa=share.copy_link>

* **2. Instagram web :**
* <https://in.docworkspace.com/d/sIOf4neHQAefvrqUG?sa=share.copy_link>

**Q43. To create HLR and TestCase on this Link**

**Q44. Write a scenario of only Whatsapp chat messages**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the All the tools are vary easy to use | Verify the it is not option to hide from particular user. |
| 2 | Verify the no advertisements on the screen to annoy the user. | Verify the not send messages without internet |
| 3 | Verify the location, images, status can be shared with friends. | Verify the profile picture is visible to every person having your contact unmark in which you are using WhatsApp |

**Q45. Write a Scenario of Pen.**

|  |  |  |
| --- | --- | --- |
| Sr .no | Test scenario(positive) | Test scenario(negative) |
| 1 | verify the types of the pan, whether it is a ball, get ink , pan, etc | verify if the pan is missing then it cannot be written properly |
| 2 | verify to check the diameter of the pan. | verify that the pan cannot be written anywhere |
| 3 | verify to check the pan can be Witten in different papers | verify if the cap is missing then ink of the pan can easily dried . |

**Q46. Write a Scenario of Pen Stand**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the pan stand is easy secure all pan’s | Verify the |
| 2 | Verify the pan stand | Verify the |
| 3 | Verify the | Verify the |

**Q47. Write a Scenario of Door.**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | verify the many typs of door like wood door, lethers door, timar door, flush door ets. | verify the door sam more expansive |
| 2 | verify the wood is material thet is easy to work. | verify the woodn door it need care and maintenance |
| 3 | verify the metal door it is door is effortless to hendle | verify thw metal door has a rusting problem |

**Q48. Write a Scenario of ATM.**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the ATM machine is they allow access to cash at any time | Verify the runs out of cash sometimes. |
| 2 | Verify the ATM machine an electronic cash providing and accepting machine | Verify the high fees in case of withdrawal from another bank’s ATM |
| 3 | Verify the minimizes transactions cost | Verify the ATM machine is limitation of cash withdrawal |

**Q49. When to used Usability Testing?**

* Usability testing is typically used during the design and development process of a product or service to evaluate its usability. It involves observing and gathering feedback from users as they interact with a prototype or a finished product to identify usability issues and make informed design decisions

=> Here are some specific scenarios when usability testing is commonly used:

1. Early design stages: Usability testing can be conducted during the initial stages of the design process to gather user feedback on concept ideas, wireframes, or low-fidelity prototypes. This helps in validating design assumptions, identifying potential usability problems, and informing iterative design improvements

2. Iterative design: Usability testing is often conducted throughout the iterative design process to validate design changes and improvements. By testing and collecting feedback on successive versions of a product, designers can refine and optimize the user experience based on user insights and preferences

3. Comparative evaluation: Usability testing can be used to compare different design options or variants. By testing multiple designs with users, it helps in identifying the strengths and weaknesses of each design option, making informed decisions about which design direction to pursue.

4. Post-launch evaluation: Usability testing can be conducted after the product is launched to gather user feedback, identify areas of improvement, and prioritize future updates or enhancements

**Q50. What is the procedure for GUI Testing?**

* GUI testing, also known as Graphical User Interface testing, focuses on evaluating the usability and functionality of the graphical elements of a software application. Here is a general procedure for conducting GUI testing
* 1.Test Planning: Define the objectives, scope, and test strategy for GUI testing. Identify the target audience and their requirements. Determine the platforms, browsers, or devices on which the application needs to be tested.

2. Identify Test Cases: Identify the test scenarios and test cases specific to GUI testing. These test cases should cover various aspects such as layout, navigation, input fields, buttons, menus, forms, error handling, and visual elements.

3. Layout and Design Testing: Evaluate the layout and design of the graphical elements. Test for consistency in color schemes, fonts, alignment, spacing, and overall visual appeal. Verify that the GUI follows established design guidelines and standards

4. Compatibility Testing: Ensure that the GUI is compatible with different browsers, screen resolutions, and devices. Test the application on various platforms and configurations to identify any rendering or functionality issues

**Q51. Write a scenario of Microwave Owen**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the cooking time is short | Verify the constraint with mela container |
| 2 | Verify the no physical change of foods | Verify the heat force control is difficult |
| 3 | Verify the melting process is easy | Verify the surface toasting is impossible |

**Q52. Write a scenario of Coffee vending Machine**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the a coffee maker allows you to brew coffee quickly and easily, | Verify the it is the very high price |
| 2 | Verify the coffee maker coms with multiple brewing options | Verify the it is need to factor that into your budget. |

**Q53. Write a scenario of chair.**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | verify the chairs is different chairs like type plastic chair, leather chair, clup chair, ets. | verify the different types of chair is moer expansive |
| 2 | verify the pastic chair are available at a low price | verify the leather chair is difficult is transport |
| 3 | verify the gaming chairs provide back support and straighten yoyr posure | verify the gleaming chair are made os leather material and ti is noy breathable |

**Q54. To Create Scenario (Positive & Negative**

**1. facebook Chat on Mobile**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the its like texting but without the cost | Verify the not send message without internet |
| 2 | Verify the no advertisements on the screen to annoy the user. | Verify the it is not option to hide from particular user. |
| 3 | Verify the location, images, status can be shared with friends. | Verify the Is not easy to maneg. |

**2. Gmail**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the gmail is a free tool | Verify the can be disruptive |
| 2 | Verify the is quick | Verify the cannot nr ignored for a long time |
| 3 | Verify the is simple | Verify the lacks a person touch |

**3. Online shopping to buy product (flipkart)**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the best way to send gifts | Verify thr no need to spend much on supporting services |
| 2 | Verify the time saving prosec | Verify the goods can be offered at lower prices |
| 3 | Verify the better options for product | Verify the delay in delivery of product |

**Q55. Write a Scenario of Wrist Watch**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the it is slim and light | Verify the lower accuracy |
| 2 | Verify the it is very accurate | Verify the most automatic watches have low power resrrve |
| 3 | Verify the dose not need a battery to function | Verify the it is ticking second hand |

**Q56. Write a Scenario of Lift (Elevator)**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the it is very faster | Verify the without use to electricity |
| 2 | Verify the applicable offishore | Verify the it is costly |
| 3 | Verify the lifting cost for high volumes generally vary low | Verify the it is most expansive |

**Q57. Write a Scenario of whatsapp Group (generate group)**

|  |  |  |
| --- | --- | --- |
| Sr.no | Test scenario(positive) | Test scenario(negative) |
| 1 | Verify the very fast group chating | Verify the it is call are limited for 8 people |
| 2 | Verify the it is | Verify the it is vary addictive |
| 3 | Verify the can creatr groups | Verify the above 100mb cannot be shared |