**Module–2(Manual Testing) Assignment**

**Q.1 What is Exploratory Testing?**

**Ans:** Exploratory Testing:

* Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution.
* It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests.

**Q.2 What is traceability matrix?**

**Ans:** 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 processshould help you keeping the virtual table up-to-date.
* Simple technique may be quite valuable (naming convention)
* 

**Q.3 What is Boundary value testing?**

**Ans:** 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 refinesequivalence partitioning.

**Q.4 What is Equivalence partitioning testing?**

**Ans:** Equivalance Paetitioning 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

**Q.5 What is Integration testing?**

**Ans**: Integration testing:

* 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.

**Q.6 What determines the level of risk?**

**Ans:** The level of risk determines by the likelihood of a risk occurring and the severity of the consequences if it does.

**Q.7 What is Alpha testing?**

**Ans**: Alpha 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.

**Q.8 What is beta testing?**

**Ans:** 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.
* It is only a kind of Black Box Testing.

**Q.9 What is component testing?**

**Ans:** Component Testing:

* A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”
* The testing of individual software components.

**Q.10 What is Functional System Testing?**

**Ans:** Functional System Testing:

* Functional Testing: Testing based on an analysis of the specification of the functionality of a component or system.
* ‘Specification’– E.g. Requirements specification, Use Cases, Functional specification or maybe undocumented.
* Functional testing verifies that each functionof the software application operates in conformance with the requirement specification.

**Q.11 What is Non-Functional Testing?**

**Ans:** 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
* Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.

**Q.12 What is GUI Testing?**

**Ans:** 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.

**Q.13 What is Adhoc testing?**

**Ans:** Adhoc Testing:

* Adhoc testing is an informal testing type with an aim to break the system.
* It does not follow any test design techniques to create test cases.
* Main aim of this testing is to find defects by random checking.
* Adhoc testing can be achieved with the testing technique called Error Guessing.

**Q.14 What is load testing?**

**Ans:** Load Testing:

* Its 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.
* Load testing is a kind of performance testing which determines a system’s performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

**Q.15 What is stress Testing?**

**Ans:** 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.
* Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions**.**

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

**Ans:** White Box Testing:

* Testing based on an analysis of theinternal structure of the component or system.
* Structure-based testingtechnique 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:

Branch Condition testing:

Branch Condition Testing requires that the True and False of each Boolean operand is tested (Boolean Operandsin this example: *If* A >30 and B >= 5)

Branch Condition Combination testing:

Branch Condition Combination Coverage would require all combinationsof Boolean operands to be evaluated

Modified Condition Decision testing:

Modified Condition Decision Coverage requires test cases to show that each Boolean operand can independently affect the outcome of the decision

Dataflow testing:

Data flow testing aims to execute sub-paths from points where each variable in a component is defined to points where it is referenced.

Linear Code Sequence And Jump (LCSAJ) testing:

LCSAJ testing requires a model of the source code which identifies control flow jumps (where control flow does not pass to a sequential statement).

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

**Ans:** Black Box Testing:

* Testing, either functional or non-functional, without reference to the internal structure of the component or system.
* Specification-based testing techniqueis also known as ‘black-box’or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
* Techniques of Black Box Testing:

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

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

Other Black Box Testing:

* Syntax or Pattern Testing

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

**Ans:** 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

**Q.19 Mention what bigbang testing is?**

**Ans:** Big Bang Testing:

* 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.

**Q.20 What is the purpose of exit criteria?**

**Ans:** 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.

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

**Ans:** Regression testing should be carried out:

* when the system is stable and the system or the environment changes.
* when testing bug-fix releases as part of the maintenance phase.

**Q.22 What is 7 key principles? Explain in detail?**

**Ans:** General Testing Principles:

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 defectsremaining in the software but, even if no defects are found, it is not a proof of correctness.
* As we find more defects, the probability of undiscovered defects remaining in a system reduces.
* However Testing cannot provethat there are nodefects present.

1. 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.
* We have learned that we cannot test everything.
* That is we must Prioritiseour testing effort using a Risk Based Approach.

1. 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 earlyas possible in the development life cycle.
* These activities should be focused on defined objectives – outlined in the Test Strategy.

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
* 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

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

6. Testing is Context Dependent:

* Testing is basically context dependent.
* Testing is done differently in different contexts
* Different kinds of sites are tested differently.
* Different industries impose different testing standards

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.
* Even after defects have been resolved it may still be unusableand/or does not fulfil the users’ needs and expectations

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

**Ans:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR No.** | **QA** | **QC** | **Tester** |
| **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. | 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 Software Test Life Cycle (STLC). | QC can be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control. |

**Q.24 Difference between Smoke and Sanity?**

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **Smoke Testing** | **Sanity Testing** |
| **1** | 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. |
| **2** | 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 with more rigorous testing |
| **3** | This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
| **4** | Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
| **5** | Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |
| **6** | Smoke testing exercises the entire system from end to end | Sanity testing exercises only the  particular component of the  entire system |
| **7** | Smoke testing is like General Health Check UP | Sanity Testing is like specialized health Check UP |

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

**Ans:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR No.** | **Criteria** | **Verification** | **Validation** |
| **1** | Defination | 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. | The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements. |
| **2** | Objective | To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements. | To ensure that the product actually meets the user’s needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment. |
| **3** | Questions | Are we building the product right? | Are we building the right product? |
| **4** | Evaluation Item | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| **5** | Activities | Reviews ∙ Walkthroughs ∙ Inspections | Testing |

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

**Ans:** Types of Performance Testing

* Load testing:

Its 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.

Stress testing is known as Endurance testing.

* Spike testing
* Volume testing
* Scalability testing

**Q.27 What is Error, Defect, Bug and Failure?**

**Ans:** 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.

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

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **Priority** | **Severity** |
| **1** | The impact of a bug on the apps functionalities | The order in which the bug must be addressed |
| **2** | Determined by app performance and functionality | Determined by business value and impact on customer |
| **3** | Refers to the technical issues needing to be addressed | Refer to the timeframe In which the issues must be addressed |
| **4** | Categories are :  Critical  Moderate  Major  Minor | Categories are:  Urgent  Medium  High  Low |
| **5** | Severity is given by QA Testers | Priority is given by Test lead or project manager |

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

**Ans**: 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’.

**Q.30 Explain the difference between Functional testing and NonFunctional testing.**

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **Functional Testing** | **NonFunctional Testing** |
| **1** | Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. | Non-Functional testingchecksthe Performance, reliability, scalability and other non-functional aspects  of the software system. |
| **2** | Functional testing is executed first | Non functional testing should be performed after functional testing |
| **3** | Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| **4** | Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| **5** | Easy to do manual testing | Tough to do manual testing |
| **6** | Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| **7** | Types of Functional testing are ∙ Unit Testing ∙ Smoke Testing ∙ Sanity Testing ∙ Integration Testing ∙ White box testing ∙ Black Box testing ∙ User Acceptance testing ∙ Regression Testing | Types of Nonfunctional testing are ∙ Performance Testing ∙ Load Testing ∙ Volume Testing ∙ Stress Testing ∙ Security Testing ∙ Installation Testing ∙ Penetration Testing ∙ Compatibility Testing ∙ Migration Testing |

**Q.31 To create HLR & TestCase of**

**(1.) Instagram only first page**

**Ans:** In Excel sheet

**(2.) Facebook Login Page : https://www.facebook.com/**

**Ans:** In Excel sheet

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

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **STLC** | **SDLC** |
| **1** | STLC is mainly related to software Testing | SDLC is mainly related to software Development |
| **2** | It is focuses only on testing the software | Besides development other phases like testing is also included |
| **3** | Goal of STLC is to complete successful testing of software | Goal of SDLC is to complete successful Development of software |
| **4** | It helps in making the software defects free | It helps in developing good quality software |
| **5** | STLC phases are performed after SDLC phases | SDLC phases are completed bofore the STLC phases |
| **6** | STLC Phases:  1.Requirement Analysis 2. Test Planning  3. Test case development  4. Test Environment setup  5. Test Execution  6. Test Cycle closure | SDLC Phases:  1.Requirements 2.Collection/Gathering  3.Analysis  4.Design  5.Implementation  6.Testing  7.Maintenance |

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

**Ans:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR No.** | **Test Scenarios** | **Test Cases** | **Test Scripts** |
| 1 | Is any functionality that can be tested | Is a set of actions executed or verify particular feature or functionality | Is a set of instruction to test an app automatically |
| 2 | Is derived from artifacts like business requirement specification and Software requirement specification | Is mostly derived from test scenarios | Is mostly derived from test cases |
| 3 | Helps test the end-to-end functionality in an agile way | Helps in exhaustive testing of an app | Helps to test specific things repeatedly |
| 4 | Is more focused on what to test | Is focused on what to test and how to test | Is focused on expected result |
| 5 | Takes late time and fewer resources to create | Requires more resources and time | Requires less time for testing but more resources for scripts creating and updating |
| 6 | Includes an end-to-end functionality to be tested | Includes test steps, data, expected result for testing | Includes different commands to develop a script |

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

**Ans:** Test Plan:

* A document describing the scope, approach, resources and schedule of intended test activities.
* Determining the scope and risks, and identifyingthe objectives of testing.
* Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.
* information that should be covered:
* **Approach**: Defining the overall approach of testing including the definition of the test levels and entry and exit criteria.
* **Integrating and coordinating the testing activities into the software life cycle activities**: acquisition, supply, development, operation and maintenance.
* **Test ware**: Defining the amount, level of detail, structure and templates for the test documentation.
* Selecting metrics for monitoring and controlling test preparation and execution, defect resolution and risk issues.
* **Process**: Setting the level of detail for test procedures in order to provide enough information to support reproducible test preparation and execution

**Q.35 What is priority?**

**Ans:** 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.

**Q.36 What is severity?**

**Ans:** 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.

**Q.37 Bug categories are…**

**Ans**: Data Quality/Database Defects:

Deals with improper handling of data in the database.

Examples:

* Values not deleted/inserted into the database properly.
* mproper/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

**Q.38 Advantage of Bugzilla.**

**Ans: Advantages:**

* Improve communication
* Increase product quality
* Improve customers satisfaction
* Ensure accountability
* Increase productivity
* Bugzilla can adapt multiple situations
* Under active development
* Constantly being put to the test by the Mozilla Foundation
* Can be installed on many operating systems, including windows, Mac and linux

**Q.39 Difference between priority and severity.**

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **Priority** | **Severity** |
| **1** | The impact of a bug on the apps functionalities | The order in which the bug must be addressed |
| **2** | Determined by app performance and functionality | Determined by business value and impact on customer |
| **3** | Refers to the technical issues needing to be addressed | Refer to the timeframe In which the issues must be addressed |
| **4** | Categories are :  Critical  Moderate  Major  Minor | Categories are:  Urgent  Medium  High  Low |
| **5** | Severity is given by QA Testers | Priority is given by Test lead or project manager |

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

**Ans:** 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 match. Scrum believes in empowering the development team and advocates working in small teams.

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.

It mainly uses physical and digital boards to allow the team members to visualize the current state of the project they are working on.

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

**Ans:**

|  |  |  |
| --- | --- | --- |
| **SR No.** | **Authorization** | **Authentication** |
| **1** | Usally Comes after Authorization | Usally first step of a security access control |
| **2** | Grants or denies permission to the user do something | Verifies the user’s identity |
| **3** | Permissions are granted and moniterd by the organisation | Common methods includes: username, Password, answer to a security question, code send via email or SMS |
| **4** | Common method include: role-based access control and attribute access control | Uses biometric data like fingerprints, Face recognition, ratinal scan |
| **5** | It’s not visible by the users | It’s visible by the users |
| **6** | Cannot be changed by users | It’s changeable by the user |

* The common problems faced in Web testing :
* Security

Cyber attacks are a major concern for web developers, and common attacks include cross-site scripting, access control, authentication breakdowns, and security misconfigurations.

* Compatibility

Different browsers interpret HTML, CSS, and JS differently, so testing for compatibility across browsers is important.

* Performance

Performance testing evaluates how well a web application scales, responds, and performs under different network conditions and user loads.

* User experience

UX testing can help identify common mistakes like a cluttered UI, too much text, or confusing calls to action.

* Functional errors

Functional testing checks that all aspects of a web application work correctly, including navigation and search functionality.

* Scalability

Real-time apps that refresh queries every few seconds can work for a few users, but may collapse under more users.

Other common problems include:

**Q.42 To create HLR & TestCase of WebBased (WhatsApp web , Instagram)**

**1. WhatsApp Web :**

**Ans:** In Excel sheel

**2. Instagram Web :**

**Ans:** In Excel sheet

**Q.43 To create HLR and TestCase on this Link. https://artoftesting.com/**

**Ans:** Skip

**Q.44 Write a scenario of only Whatsapp chat messages.**

**Ans:** Scenario of only Whatsapp chat messages:

* Verify that the user can set a chat wallpaper.
* Verify that the user sets privacy settings like turning on/off last seen, online status, read receipts, etc.
* Verify that the user can update notification settings like – notification sound, on/off, and show preview for both group and individual chats.
* Verify that the user can take the complete chat backup of his chats.
* Verify that the user can update the phone number that is used by the WhatsApp application.
* Verify that the user can disable/delete his Whatsapp account.
* Verify that the user can check data usage by images, audio, video, and documents in WhatsApp chats.

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

**Ans:** Scenario of Pen:

* Verify that the length and the diameter of the pen are as per the specifications.
* Verify the outer body material of the pen. Check if it is metallic, plastic, or any other material specified in the requirement specifications.
* Check the color of the outer body of the pen. It should be as per the specifications.
* Verify that the brand name and/or logo of the company creating the pen should be clearly visible.
* Verify that any information displayed on the pen should be legible and clearly visible.
* 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.
* Verify the color of the ink on the pen.
* Check the odor of the pen’s ink on writing over a surface.
* Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
* Verify that the text written by the pen should have consistent ink flow without leaving any blob.
* Check that the pen’s ink should not leak in case it is tilted upside down.
* Verify if the pen’s ink should not leak at higher altitudes.

**Q.46 Write a Scenario of Pen Stand.**

**Ans:** Scenario of Pen Stand:

* Verify that the length and the diameter of the pen stand are as per the specifications.
* Check the weight of the pen stand. It should be as per the specifications.
* Verify if the pen stand is with in a form of opened or Pack.
* Verify the colour of the pen stand.
* Verify the Material of pen stand, it be in wooden, metal, glass, plastic or any else.
* Verify that how many pens can handle in stands.
* Verify that can keep other stationaries in pen stand.
* Verify that the design of the pen stand is like to use of office, home, schools or other.
* Verify that the pen stand can have a single holder or multiple holder.
* Verify that the surface of pen stand is flat.
* Verify that the pen stand design can hold how much weight can handle.

**Q.47 Write a Scenario of Door.**

**Ans:** Scenario of Door:

* Verify if the door is single door or bi-folded door.
* Check if the door opens inwards or outwards.
* Verify that the dimension of the doors are as per the specifications.
* 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.

**Q.48 Write a Scenario of ATM.**

**Ans:** 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 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.
* Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly.
* 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 is 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 presented with different account type options like- saving, current, etc.
* Check that in case the ATM machine runs out of money, a proper message is displayed to the user.

**Q.49 When to used Usability Testing?**

**Ans:** Importance of use of 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
* Usability Testing identifies usability errors in the development cycle and can save a product from failure.

**Q.50 What is the procedure for GUI Testing?**

**Ans:** Procedure for GUI Testing:

* Build the model
* Determine Inputs for the model
* Calculate expected output for the model
* Run the Tests
* Compare the actual output with the expected output
* Decision on further action on the model
* Some of the modeling techniques from which test cases can be derived:
* Charts – Depicts the state of a system and checks the state after some input.
* Decision Tables – Tables used to determine results for each input applied
* Model based Testing is an evolving technique for the generating the test cases from the requirements.
* Its main advantage, compared to above two methods, is that it can determine undesirable states that your GUI can attain.

**Q.51** **Write a scenario of Microwave Owen.**

**Ans:** Scenario of Microwave Owen:

* 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 the oven heats food at the desired temperature within a specified time duration.
* Verify the ovens functioning with the 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 function’s 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 the oven’s functionality with different kinds of food – solid, and 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.

**Q.52 Write a scenario of Coffee vending Machine.**

**Ans:** Scenario of coffee vending machine:

* Verify that the dimension of the coffee machine is as per the specification.
* Verify that outer body, as well as inner part’s material, is as per the specification.
* Verify that the machine’s body color as well brand is correctly visible and as per specification.
* Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.
* Verify that the quantity of hot water, milk, coffee powder per serving is correct.
* Verify the power/voltage requirements of the machine.
* Verify that coffee should not leak when not in operation.
* Verify the amount of coffee served in single-serving is as per specification.
* Verify that the digital display displays correct information.
* Check if the machine can be switched on and off using the power buttons.
* Check for the indicator lights when the machine is switched on-off.
* Verify that each button has an image/text with it, indicating the task it performs.
* Verify that pressing the coffee button multiple times leads to multiple serving of coffee.

**Q.53 Write a scenario of chair.**

**Ans:** 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.

**Q.54 To Create Scenario (Positive & Negative)**

**(1) Gmail (Receiving Mail)**

* **Positive:**
* 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.
* **Negative:**
* Verify that user can not able to receive the mail without sender name.
* Verify that when user read the email counting is not decreases.
* Verify that when user receives the mail it can not shows in inbox .
* Verify that user can not able to start the emails.

**(2) Online shopping to buy product (flipkart)**

**Ans:** scenario of online shopping to buy product:

* **Positive:**
* Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.
* Verify that the user can add to the cart one or more products.
* Verify that users can add products to the wishlist.
* Verify that the user can see the previously added products on the cart page, after signing in to the application.
* Verify that the user can successfully buy more than one products that were added to his/her cart.
* Verify that the user cannot add more than the available inventory of the product.
* 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.
* Verify that the delivery can be declined during checkout for the places where shipping is not available.
* Verify that the Cash on Delivery option of payment is working fine.
* Verify that the different prepaid methods of payments are working fine.
* Verify that product return functionality works correctly.
* **Negative:**
* Verify that user can’t buy product without enter address.
* Verify that user can’t buy a product without select payment method
* Verify that user can’t buy a product with add quality more that it’s availability.
* Verify that user can buy a product without rating that product.
* Verify that user can buy a product without add flipkart gift card
* Verify that user can’t able to buy damage product.

**Q.55 Write a Scenario of Wrist Watch.**

**Ans:** 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 its 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.

**Q.56 Write a Scenario of Lift(Elevator).**

**Ans:** Scenario Of Lift:

* 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 the lift moves to the particular floor as the button of the floor is clicked.
* Verify that the lift stops when the up/down buttons on a particular floor are pressed.
* Verify if there is an emergency button to contact officials in case of any mishap.
* Verify the time duration for which the door remains open by default.
* Verify if the lift interior is having proper air ventilation.
* Verify lighting in the lift.
* Verify that at no point the lift 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 of capacity limit is reached users are prompted with a warning alert- audio/visual.

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

**Ans:** Scenario of Whatsapp Group(Genetare group):

* Verify that a user can successfully create a new group.
* Confirm that the group is immediately visible in the user's chat list.
* Test adding multiple members to the group.
* Confirm that all added members receive an invitation and join successfully.
* Test adding a group description.
* Test the ability to delete a group, ensuring it is removed from all members’ accounts.
* Verify that the admin can initiate group deletion process after removing all the members.
* Check if a member can exit and delete the group for only themselves.
* Verify that members receive notifications when removed from a group.
* Test any recovery options available after group deletion, such as restoring archived content.
* Verify that the admin role can be assigned to a specific group member.
* Test the functionality of admin-specific permissions, such as adding/removing members and changing group settings.
* Check if admins receive notifications for important group activities, like member additions or removals.
* Verify that only admins can create, add/remove, and make other users admin in the group.
* est the ability to delete a group, ensuring it is removed from all members’ accounts.
* Verify that the admin can initiate group deletion process after removing all the members.
* Check if a member can exit and delete the group for only themselves.
* Verify that members receive notifications when removed from a group.
* Test any recovery options available after group deletion, such as restoring archived content.

**Q.58 Write a Scenario of Whatsapp payment.**

**Ans:** Scenario of Whatsapp payment:

* Payment status: Check for a callback from WhatsApp with details about the payment status.
* Order status: Verify that the order status message is correct and matches the payment IDs from the PG merchant website.
* Payment options: Ensure that all payment options, such as UPI, credit/debit card, and netbanking, are working properly.
* Error messages: Check for error messages if a payment fails.
* Language and currency: Verify that the language and currency are relevant to the location.
* Transaction response: Check the application's response after a transaction.
* Demo credit/debit cards: Arrange demo credit/debit card numbers for testing.
* Transaction messages: Check the messages displayed after a successful or unsuccessful transaction.
* Redirect: Verify that the user is redirected to the correct webpage after a successful transaction.