Module 2 Manual Testing

**1) What is integration testing?**

* **Integration testing** is a software testing phase that verifies how components of a software application work together.
* Integration testing checks connection between 2 modules works properly.
* Example, after registration data will be stored in database so log in data from registration data table should be fetch and logged into system

**2) What is functional testing?**

* **Functional testing** is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user’s expectations.
* Example : Integration testing, Unit testing, smoke testing , regression testing, sanity testing etc,.

**3) What is Non-Functional Testing?**

* **Non-Functional Testing** is a type of software testing that focuses on evaluating the non-functional aspects of a system, such as its performance, usability, reliability, and security. It complements functional testing by assessing how well the system operates rather than just whether it operates correctly.
* Example: Usability testing, performance testing, volume testing, security testing, black box testing

**4) What is load testing?**

* Load Testing is a type of performance testing that evaluates a system's behaviour under expected workload conditions. It aims to determine how the system responds when multiple users access it concurrently.

**5) What is stress testing?**

* **Stress Testing** is a non-functional testing technique that evaluates a system's behaviour under extreme or peak load conditions. It's designed to push the system beyond its normal operating limits to identify its breaking point and assess its stability and reliability under stress.

**6) What is white box testing and list the types of white box testing?**

* **White Box Testing** is a software testing method where the tester has access to the internal structure and workings of the application. This knowledge is used to design test cases that can verify the correctness of the software at the code level.
* White box testing is primarily performed by **software developers** who have a deep understanding of the code and its internal structure.
* **Types of White Box Testing:**

1. **Statement Coverage:** Ensures that every line of code is executed at least once during testing.
2. **Branch Coverage:** Ensures that every branch of a conditional statement is executed at least once.
3. **Path Coverage:** Ensures that every possible path through the code is executed at least once.
4. **Condition Coverage:** Ensures that both true and false conditions of each condition in the code are tested.
5. **Decision Coverage:** Ensures that all possible outcomes of each decision point in the code are tested.

**7) What is black box testing? What are the different black box testing techniques?**

* **Black Box Testing** is a software testing method where the tester examines the functionality of an application without peering into its internal structures or workings. It's like treating the software as a "black box" – you know what goes in (input) and what comes out (output), but not how it works inside.
* Black box testing is done by **software testers**, who are professionals specifically trained in testing methodologies. They design and execute test cases based on the software's requirements and specifications.
* **Key Characteristics:**
* **Focus on Functionality:** Black box testing primarily focuses on verifying that the software performs its intended functions correctly.
* **No Internal Knowledge Required:** Testers don't need to know the internal code or design of the software.
* **Based on Requirements:** Test cases are designed based on the software's requirements and specifications.
* **User Perspective:** Black box testing often simulates user behavior to identify issues that might not be apparent to developers.

**8) Difference between QA v/s QC v/s Tester**

|  |  |  |
| --- | --- | --- |
| **QA(Quality assurance)** | **QC(Quality Control)** | **Tester** |
| A broader process that focuses on preventing defects by establishing and implementing processes and standards throughout the entire software development lifecycle. | A narrower process that focuses on identifying and correcting defects in the final product. | A specific role within the QC process responsible for executing tests, analyzing results, and reporting defects. |
| QA establishes the quality framework. | QC verifies the product against that framework. | Testers are the individuals who perform the testing activities within the QC process. |
| **QA** would establish the overall quality framework, such as coding standards, testing methodologies, and risk management processes. | **QC** would focus on the actual testing activities, such as functional, performance, and security testing, to identify and address defects. | **Testers** would be the individuals responsible for designing and executing specific test cases, logging bugs, and verifying bug fixes. |

9) **Difference between Smoke and Sanity?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Smoke | | Sanity | |
| Goal | | **Goal:** To quickly verify the stability and basic functionality of a new build or after a major code change. | | **Goal:** To verify that the specific functionality or component affected by a recent code change or bug fix is working as expected. | |
| Scope | | **Scope:** Covers critical functionalities and core pathways of the entire system. | | **Scope:** Narrower, focuses on specific areas where changes have been made. | |
| Depth | | **Depth:** Shallow, focuses on ensuring the system is stable enough for further testing. | | **Depth:** More detailed than smoke testing, but less comprehensive than regression testing. | |
| Timing | | **Timing:** Performed early in the testing cycle, often after each build. | | **Timing:** Performed after smoke testing, often triggered by code changes or bug fixes. | |

**10) Difference between verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| The process of checking whether the software is being built **correctly** according to the specifications. | The process of checking whether the software being built is the **right** product. |
| Ensures the software conforms to the specified requirements and design | Ensures the software meets the user's needs and expectations. |
| Reviews, inspections, walkthroughs, desk-checking. | Functional testing, system testing, integration testing, user acceptance testing. |
| Occurs throughout the development lifecycle, often before validation. | Typically occurs after verification, towards the end of the development cycle. |
| To find defects early in the development process. | To ensure the software meets user requirements and expectations. |
|  |  |

**11) What is error, defect, bug and failure?**

**1. Error**

* **Definition:** A mistake, misconception, or misunderstanding on the part of the developer. It can arise from incorrect assumptions, misinterpretations of requirements, or simply human error during the coding process.
* **Example:** Developer made a mistake in coding

**2. Defect**

* **Definition:** A deviation from the expected behavior of the software. It's a flaw in the software that causes it to behave incorrectly.
* **Example:** When software have to send same output but didn’t get output and program stop execute from that point

**3. Bug**

* **Definition:** An informal term often used interchangeably with "defect." It generally refers to any unexpected behavior or problem in the software.
* **Example:** when user enters wrong password but system log in from that it is bug

**4. Failure**

* **Definition:** The inability of the software to perform its intended function. It's the observable consequence of a defect.
* **Example:** If a critical bug prevents the software from starting or processing data correctly, it results in a system failure.

**12)** **Difference between Priority and Severity**

|  |  |  |
| --- | --- | --- |
|  | Priority | Severity |
| Definition | Describes the technical impact of the defect on the software. | Determines the order in which defects should be fixed. |
| Focus | How severely the defect affects the software's functionality, stability, or security. | How quickly the defect needs to be fixed to minimize business impact or customer dissatisfaction. |
| Assessment | Objective, based on technical criteria. | Subjective, based on business needs and urgency. |
| Examples | Critical, Major, Minor, Cosmetic | High, Medium, Low |
| Relationship | Often influences priority. A critical severity defect is usually high priority. | May not always directly correlate with severity. A minor cosmetic issue might be high priority if it significantly impacts user experience. |

**13)** **Write a Scenario of Pen**

* 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.
* Verify if the pen is with a cap or without a cap.
* Verify the colour of the ink on the pen.
* Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
* 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.
* Verify if the text written by the pen is erasable or not.
* Check the functioning of the pen by applying normal pressure during writing.
* Verify that text written by pen should not get faded before a certain time as mentioned in the specification.
* Verify that the user is able to write normally by tilting the pen at a certain angle instead of keeping it straight while writing.
* Check the grip of the pen, and whether it provides adequate friction for the user to comfortably grip the pen.
* In the case of a ballpoint pen, verify the size of the tip.
* In the case of a ball and gel pen, verify that the user can change the refill of the pen easily.

**14) Write a Scenario of Pen Stand**

* Verify which material used in pen stand
* Verify how much pressure pen stand endure when pen is put
* Verify size and shape of pen stand
* Verify How much weight penstand ensure
* Load the stand with the maximum numbers of pens specified by manufacturer
* Expose the stand to extreme temperatures like too hot or too cold
* Expose pen stand in high humidity to check for any adverse effects in the material
* Verify penstand can stand on different surfaces like smooth, uneven, inclined
* Apply slight force to stand to access its resistance to movement

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

**16)Write a Scenario of ATM**

* Verify the ‘ATM Card Insertion Slot’ is as per the specification
* Verify the ATM machine accepts card and PIN details
* Verify the error message by inserting a card incorrectly
* Verify the error message by entering an incorrect PIN
* Verify that the user is asked to enter the PIN after inserting a valid ATM Card
* Verify that PIN is encrypted
* Verify that there is an action like blocking of card occurs when the total no. of incorrect PIN attempts get surpassed
* Verify the machine logs out of the user session immediately after successful withdrawal
* Verify the message when there is no money in the ATM
* Verify the language selection functionality
* Verify the cash withdrawal functionality by entering some valid amount
* Verify the cash withdrawal functionality by entering an amount less than 100 or greater than given amount
* Verify the cash withdrawal functionality by entering an amount greater than per day limit
* Verify the user is allowed to enter the amount again in case the amount entered is not valid. A proper message should be displayed.
* Verify the ATM machine successfully takes out the money.
* Verify the ATM machine takes out the balance printout after the withdrawal
* Verify the font of the text displayed in ATM screen
* Verify the functionality of all the buttons on the keypad
* Verify the text on the buttons visible clearly.
* Verify that touch of the ATM screen is smooth and operational
* Verify the user is allowed to choose different account types like Savings, Current etc.,
* Verify the functionality of the cash dispenser
* Verify the functionality of the receipt printer
* Verify whether the printed data is correct or not in the receipt
* Verify how much time the system takes to log out.

**19) Write a 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 minimum or maximum 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.

**21) Write a scenario of chair**

* Verify that the chair is stable enough to take an average human load
* Check the material used in making the chair-wood, plastic etc
* Check if the chair’s leg are level to the floor.
* Check the usability of the chair as an office chair, normal household chair.
* Check if there is back support in the chair.
* Check if there is support for hands in the chair.
* Verify the paint’s type and colour.
* 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