**What is exploratory Testing?**

Exploratory testing is a concurrent process where

* Test design, execution and logging happen simultaneously.
* Testing is often not recorded.
* Makes use of experience, heuristics and test patterns.
* Testing is based on a test charter that may include
* Scope of the testing (in and out)
* The focus of exploratory testing is more on testing as a “thinking” activity.
* A brief description of how tests will be performed.
* Expected problems.
* Is carried out in time boxed intervals.

**What is traceability matrix?**

Test conditions should be able to be linked to their sources in the test basis is known as traceability.

* **Type of Traceability matrix**
* **Forward Traceability:** Mapping of requirements to test cases.
* **Backward Traceability:** Mapping of test cases to requirements.
* **Bi-directional Traceability:** mapping requirements to test cases, and test cases to requirements.
* **Pros of traceability matrix:**
* Make obvious to the client that the software is being developed as per the requirement.
* Make sure that all requirements included in test cases.
* Easy to identify the missing functionalities.
* **Cons of Traceability:**
* Poor or unknown test coverage, more defects found in production.
* It will lead to miss some bugs in earlier test cycle which may arise in later test cycle. Then a lot of discussion arguments with other teams and managers before release.

**What is ad-hoc testing?**

* Ad-hoc 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.
* It does not create test cases altogether.
* This testing is primarily performed if the knowledge of tester in the system under test is very high.
* Tester randomly tests the application without any test cases or any business requirement document.
* Ad-hoc testing does not follow any structured way of testing and it is randomly done on any part of application.
* Main aim of this testing is to find defects by random checking.
* Ad-hoc testing can be achieved with the testing technique called error guessing.
* Error guessing can be done by the people having enough experience on the system to “guess” the most likely source of errors.

**Types of ad-hoc testing**

**Buddy testing:** Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the tester develop better test cases and development team can also make design changes early. This testing usually happens after unit testing completion.

**Pair testing:** Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the test and another person can take notes on the finding. Role of the persons can be a tester and scriber during testing.

**Monkey testing:** Randomly test the product or application without test cases with a goal to break the system.

**What Is Alpha testing?**

* It is always performed by the developers at the software development site.
* Sometimes it is also performed by independent testing team.
* Alpha testing is not open to market and public.
* It is conducted for the software application and project.
* It is always performed in virtual environment.
* It is always performed within the organization.
* It is the form of acceptance testing.
* Alpha testing is definitely performed and carried out at the developing organizations location with the involvement of developers.
* It comes under the category of both white box and black box testing.

**What is Beta testing?**

* It is always performed by the customer at their own site.
* It is not performed by independent testing team.
* It is always open to market and public.
* It is usually conducted for software product.
* It is performed in real time environment.
* It is always performed outside the organization.
* It is also the form of acceptance testing.
* It is only a kind of black box testing.

**What are the categories of Defects?**

* **Security defect:** 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. Ex.
* **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. Ex.
* Improper error/warning/UI messages.
* Spelling mistakes.
* Alignment problems.

**What is GUI testing?**

* Check all the GUI elements for size, position, width, length and acceptance of characters and numbers for instance; you must be able to provide inputs to the input field.
* Check you can execute the intended functionality of the application using the GUI.
* Check error message and displayed correctly.
* Check for clear demarcation of different sections on screen.
* Check font used in application is readable.
* Check the alignment of the text is proper.
* Check the color of the font and warning message is aesthetically pleasing.
* Check that the images have good clarity and are properly aligned.
* Check the positioning of GUI elements for different screen solution.

**Approaches of GUI testing:**

* Manual based testing.
* Under this approach graphical screens are checked manually by testers in conformance with requirements document.
* Record and replay.
* GUI testing can be done using automation tools. This is done in two parts; during record, test steps are captured into automation tool, during playback the recorded test steps are executed on the application under test. Ex. QTP.
* Model based testing.
* A model is graphical description.

**What is Stress Testing?**

* Stress testing is done to make sure that the system would not crash under crunch situations.
* Stress testing is also known as endurance testing.

**Need of stress testing**

* During festival time, an online shopping site may witness a spike in traffic, or when it announces a sale.
* When a blog is mentioned in a leading newspaper, it experiences a sudden surge in traffic.
* To check whether the system works under abnormal conditions.
* Displaying appropriate error message when the system is under stress.

**Goal of stress testing**

* The goal of stress testing is to analyze the behavior of the system after failure. For stress testing to be successful, system should display appropriate error message while it is under extreme conditions.
* To conduct stress testing, sometimes, massive data sets may be used which may get lost during stress testing.
* Tester should not lose this security related data while doing stress testing.
* The main purpose of stress testing is to make sure that the system recovers after failure which is called as recoverability.

**What is Load Testing?**

* Load testing is to test the system behavior under normal workload conditions, and it is just testing or simulating with the actual workload.
* Load testing identifies the bottlenecks breaking, the system under various workloads and checks how the system reacts when the load is gradually increased.
* Load testing does not break the system.

**Need of Load Testing:**

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

**Example:** An airline website was not able to handle 10,000+ users during a festival time.

Face book.

**Goals of Load Testing:**

Load Testing Identifies the following Problems:

* Response time for each transaction.
* Performance of system components under various loads.
* Performance of database components under different loads.
* Network delay between the client and the server.
* Software design issues.

**Strategies of Load Testing:**

* Manual Load Testing.
* In house (Organization) developed load testing tools.
* Open source load testing tools.
* Enterprise (Record and play) load testing tools.

**What is Functional System Testing?**

* Testing based on an analysis of the specification of the functionality of 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.
* Each and every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results.
* This testing involves checking of user interface, API**s,** database, security, client/server applications and functionality of the application under test.

**Non-functional Testing?**

* Testing the attributes of component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability, and portability.
* May be performed at all test levels (not just non-functional system testing).
* Non functional testing includes, but not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing.
* It is the testing of ‘how’ the system works.

**What are Error, Defect, Bug and Failure?**

**Error:** A mistake in coding is called error. A discrepancy between a computed observed or measured value or condition and the true specified, the theoretically correct value or condition. This can be misunderstanding of the internal state of the software, an oversight in terms of memory management confusion about the power way to calculate a value etc.

**Defect:** Error found by tester is called defect. Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

**Bug:** Defect accepted by the development team then it is called bug. A fault in a program which causes the program to perform in an unintended or unanticipated manner. Bug is terminology of tester.

**Failure:** Build does not meet the requirement then it is failure. The inability of a system or component to program its required functions within specified performance requirement. see:- Bug, crash, exception and fault.

**What is 7 key principal? Explain in detail?**

1. **Testing shows presence of the defects:** Testing can show that defect are present but can not prove that there are no defects. Testing reduces the probability of undiscovered defect.
2. **Exhaust testing is impossible:** Testing everything including all combination of inputs and preconditions is not possible. So instead of doing the exhaustive testing we can use risks and priorities to focus testing effort.
3. **Early testing:** Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
4. **Defect clustering:** A small number of modules contain most of the defect discovered during pre-release testing or are responsible for the most optional failures.
5. **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 these “pesticide paradox” the test cases need to be regularly reviewed and revised and new different tests need to be written to exercise different parts of the software or system to potentially find more defects.
6. **Testing is contexts dependent:** Different kinds of sites are tested differently. For example critical software is tested differently from an e-commerce site.
7. **Absence of error fallacy:** If the system built is unusable and does not fulfill the user needs and expectations then finding and fixing defects does not help.

**What is black box testing? List the type of black box testing.**

* The test engineers perform the black box testing.
* What the software is supposed to do but is not aware of how it does it.
* To perform black box testing, there is no need to have an understanding of the programming languages.
* In this, we will verify the functionality of the application based on the requirement specification.
* In this there is no need to know about the internal design of the code.
* Test design techniques: Decision table testing, all pairs testing, Equivalence partitioning, Boundary value analysis, cause effect glass.
* Can be applied virtually to every level of software testing: unit, integration, system and acceptance.

**Types of Black Box Testing**

**Equivalence Partitioning:**

* Aim is to treat groups of inputs as equivalent and to select one representative input to test the mall.
* It can be used for all levels of testing.
* It is the process of defining the optimum number of test 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.

**Boundary Value Analysis:**

* Boundary value analysis is a methodology for designing test cases that concentrates software testing effort on cases near the limits of valid range, boundary value analysis is a method which refines equivalence.

**Decision Table:**

* The techniques of equivalence partitioning and boundary value analysis are often applied to specific situations or inputs.
* Based software testing techniques, decision tables and state transition testing are more focused on business logic or business rules.
* This technique is sometimes also referred to as a cause effect table.

**State Transaction Testing:**

* A transition between two states of a component or system.
* A black box test design technique in which test cases are designed to execute valid and invalid state transitions. Also known as N-switch testing.

**What is White Box Testing?**

* White box testing (also known as clear box testing, glass box testing, and transparent box testing and structural testing) is a method of software testing that tests internal structures or working of an application.
* White box testing can be applied at the unit, integration and system levels of the software testing process.
* Here are some of the top white box testing tools to use: varicose, Capp unit, N unit, RCUNIT etc.

**What is Big Bang Testing?**

In a big-bang approach, first all the modules are individually tested. Next all modules are put together to construct the entire system which is tested as a whole. Sometimes developers use the big-bang approach to integrate small system. However, for large system, this approach is not recommended for the following reasons:

* In a system with a large number of modules, there may be many interface defects. It is difficult to determine whether or not the cause of a failure is due ti interface errors in a large and complex system.
* In a large system, the presence of a large number of interface errors is not an unlikely scenario in software development.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **QA** | **QC** | **TESTING** |
| 1. | Activities which ensure the implementation of process, procedures and standard 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/defect in software. |
| 2. | Focuses on process and procedure rather than conducting actual testing on the system. | Focuses on actual testing by executing software with intended 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. | Corrective process. | Preventive process. |
| 5. | It is subset of software test life cycle (STLC). | QC can be considered as the subset of quality assurance. | Testing is the subset of quality control. |

**Difference between Verification and Validation?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Verification** | **Validation** |
| 1. | Are we building it Right thing. | Have you built the right thing? |
| 2. | Verification is the static testing | Validation is dynamic testing |
| 3. | It does not include the execution of the code. | It includes the execution of the code. |
| 4. | Methods used in verification are reviews, walkthroughs, inspection and desk-checking. | Methods used in validation are black box testing, white box testing and non-functional testing. |
| 5. | It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. |
| 6. | It finds the bugs in the early stages of the development. | It can only find the bugs that could not be found by the verification process. |
| 7. | Quality assurance team does verification. | Validation is executed on software code with the help of testing team. |
| 8. | It comes before validation. | It comes after verification. |

**What is Boundary Value Testing?**

The black box testing techniques are helpful for detecting any error or threats that happened at the boundary values of valid or invalid partitions rather than focusing on the center of the input data.

**What is Equivalent Partitioning?**

* A black box technique which can be applied to all levels of testing.
* Divide set of test conditions into partitions that can be considered the same.

**What is Integration Testing?**

Testing performed to expose defects in the interfaces and in the interactions between integrated components or system. See also component integration testing, system integration testing.

**What is Component Testing?**

Testing performed to expose defects in the interfaces and interaction between integrated components.

**What determines the level of Risk?**

* A factor that could result in future negative consequences.
* Process improvement can prevent those defects re-occurring.
* When testing finds defects, the quality of software system increases when those defects.

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

* **Data quality/Database defect:** Deals with improper handling of data in the database.

**Examples:**

* Values not detected/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 critical functionality of the application. **Examples**: Exceptions.
* **Functionality Defects:** The defect affects the functionality of the application.

**Examples:**

* All Java Script 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 through permission not given.
* **User Interface Defects:** As the name suggest, the bugs deal with problems related to UI are usually considered less severe.

**When should “Regression Testing” be performed?**

* Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.
* If the test is re-run and passes you can not necessarily say the fault has been resolved because, you also need to ensure that the modification have not caused unintended side-effects elsewhere and that the modified system still meets its requirements. Regression testing.

**What is the purpose of exit criteria?**

* The purpose of exit criteria is to define when we STOP testing either at the:
* End of all testing: i.e. product Go live.
* End of phase of testing (e.g. hand over from system test to UAT).

**Difference between smoke v/s sanity testing?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Smoke** | **Sanity** |
| 1. | It is done at the early stage before regression testing. | The goal of sanity testing is to determine that the proposed functionalities are working roughly as expected. |
| 2. | The purpose of smoke testing is to reject badly performing application so that the QA team does not have to waste time in the installation or testing of a software application. | Testing performed after receiving a software build with little changes in code or functionality to ascertain that certain bugs have been fixed in advanced to resolve workflow issues. |
| 3. | The objective is clear here to verify the most critical functionalities of the system either they are working fine or not. | If sanity testing fails then the build is rejected directly to save time and costs that are involved in more rigorous testing. |

**What are difference between Severity and Priority?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Severity** | **Priority** |
| 1. | Based on bug itself. | Based on the expected bug impact. |
| 2. | Intrinsic quality of the bug. | Extrinsic quality of the bug. |
| 3. | Set by the tester. | Set by the developer. |
| 4. | Depends on a single bug. | May depends on priority of other open bugs. |
| 5. | Typically static. | Typically dynamic. |
| 6. | Has only few types i.e. Critical, High, Medium, Low and enhancement. | Has only types e.g. P1 (Immediate) may be broken into P11, P12 and P13. |

**What is the difference between STLC and SDLC?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **SDLC** | **STLC** |
| 1. | SDLC is a development life cycle. Mainly related to development. | STLC is a testing life cycle. Mainly related to testing. |
| 2. | In SDLC, phases includes requirement gathering, Analysis, Design, coding, Testing, Development. | In STLC, phases includes requirement analysis, Test planning, Test design, Environment setup, Test Execution, Test closure. |
| 3. | In SDLC process, in requirement gathering phase, BA will gather requirements and developer will perform the development. | In STLC process, the testing team will check the software from functional or non-functional point of view. |
| 4. | SDLC is intended for the journey of successful software development. | STLC is needed to find any pitfalls or shortcomings in the testing phase. |
| 5. | In SDLC the development team creates the high and low level design plans. | In STLC the test analyst creates the system and integration Test plan. |

**Explain what test plan is? What is the information that should be covered?**

* A document describing the scope, approach, resources and schedule of intended test activities.
* As more information becomes available.
* As new risks arise or others are mitigated.
* Not set in concrete, but changes must be carefully managed.
* It making a decision about:
* What to test.
* How to test.
* When and how the test activities should be done and when they be stopped.
* How the test result will be evaluated.

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

**Test Scenario:**

* A test scenario is any functionality that can be tested. It is also called test condition, or test possibility.
* Test scenario is “what to be tested”.
* Test scenario is nothing but test procedure.
* The scenarios are derived by use-cases.

**Test Script:**

* 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.
* There are various means for executing test scripts.
  + Manual Testing.
  + Automation Testing.

**Test Case:**

* Test case consist of a set of input values, execution precondition, expected results and executed post condition, developed to cover certain test condition.
* Test case is a set of variable under which a tester will determine if a requirement upon an application is partially or fully satisfied.

**What is Priority?**

Defect priority relative and business focused priority defines the order in which we should resolve a defect. **Ex:** if a company name is misspelled in the homepage of the website, then the priority is high and severity is low.

**Priority can be following types:**

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

**Medium:** The defect should be resolve in the normal course of development activities. It can be waiting until a new build or version created.

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

**Critical:** Extremely urgent, resolve immediately.

**What is Severity?**

Severity is absolute and customer focused it is extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system. **Ex:** If an application or web page crashes when a remote link 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.

**Bug categories are:** Security, Database, Functionality (critical/general) UI.

**Write a scenario of Pen?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Test Scenario Positive** | **Test Scenario Negative** |
| 1. | Verify the type of pen whether it is ball point, gel point or ink pen. | If the pen cap is missing, so ink will be dry, so cannot write properly. |
| 2. | Verify the paper of the pen whether should I write properly. | Verify the pen grip is broken then cannot hold the pen properly. |
| 3. | Verify the material of the pen whether it is plastic, fiber ,metal etc. | Verify the pen cap stand us broken then cannot put in pocket with standby |
| 4. | Verify the pen width which is suitable for hold. | If pen ink is empty so every time to change refill. |
| 5. | Verify the height of the pen which is again for hold. | If sometime ink is empty so we can not changed the refill. |
| 6. | Verify the refill color whether it is blue black red etc. | Verify if pen is not waterproof, so that we cannot write on paper. |
| 7. | Verify the leakage of ink whether it is leaking or not. | Verify if we give pressure to the pen on paper, so it put holes on the paper. |
| 8. | Verify the pen cap holder which is proper or not. | Verify the pen refill is empty, so we cannot pour the ink in refill. |
| 9. | Verify the click pen spring which is placed inside the pen is not broken. | Verify the pen tip is uneven, so that we cannot write anything. |
| 10. | Verify the pen mechanism, whether it is working properly or not. | Verify if the ink is spread, so that it will mess your clothes. |
| 11. | Verify the fountain pen refill which is broad or not. | Writing. |

**Write a scenario of Door?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.N.** | | **Test Scenario Positive** | | **Test Scenario Negative** |
| 1. | | Check the dimension of the door which is suitable for everyone. | | Problem of mechanism. |
| 2. | | Verify the materials of the door whether is glass, metal, wooden, etc. | | Doesn’t fit to close. |
| 3. | | Verify the quality of the door whether it is strong or not. | | Verify if the door is broken somewhere then it may leads to hurt someone. |
| 4. | | Check the design of the door whether it is floral, plain, etc. | | Verify the cost of the door. |
| 5. | | Check the color of the door which is according to the person. | | Verify the wood door is nearby the fire, so that it catches the fire easily. |
| 6. | | Verify the handle of the door whether it is fixed to the door properly. | | Verify if we place the hand unfortunate in between the lock we may hurt. |
| 7. | | Verify the auto closer door mechanism which is working or not. | | Verify if the glass door is not handled properly it may have the chance of broken. |
| 8. | Verify the magnet holder of door which catches the door or not. | | Slowly. | |
| 9. | Verify the door stopper which stops the door properly or not. | | Moisture. | |
| 10. | Verify the door stopper rubber whether it is broken or not. | | Verify if the door has a no guarantee so that it may loosen the door installation. | |
| 11. | Verify the peek holes of the door which is properly working or not. | | Verify if the door is heavy, so that we cannot handle properly. | |

**Write a scenario of ATM?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive Test Scenario** | **Negative Test Scenario** |
| 1. | Verify the power backup should be present at ATM. | Verify the functionality by entering a wrong pin number for a particular number of times. |
| 2. | Verify that card reader should be present. | Verify the card ATM machine functionalities by inserting an expired card. |
| 3. | Verify that receipt printer should be present and working. | Verify the cash withdrawn functionality by entering invalid amount such as 10, 20, 50 |
| 4. | Verify the cash dispenser is working as expected. | Verify the ATM machine functionalities by entering amount greater than available balance. |
| 5. | Verify that the keypad should be working and covered. | Verify the ATM machine functionalities by entering amount greater than per day and per transaction limit. |
| 6. | Verify that buttons are displayed on screen of ATM machine. | Verify the print receipt text is properly displayed. |
| 7. | Verify the font of text on the screen, it should be clearly visible. | Machine finds wrong pin. |
| 8. | Verify that when card inserted in ATM, pin should be asked from user. | Machine takes 3 invalid attempts. |
| 9. | Verify that when user enters correct pin, the user details should be displayed on ATM. | Machine cannot logout client session after withdrawal. |
| 10. | Verify that ATM machine asks to user for the amount to be withdrawn. | Machine does not accept either visa or master-card or both debit/credit cards if reader is not working. |
| 11. | Verify that if user enters amount greater than daily withdrawn limit, error message is displayed. | Machine does not print the withdrawal amount. |

**Write a scenario of only whatsapp chat message?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Test Scenario Positive** | **Test Scenario Negative** |
| 1. | Verify the user can download/install whatsapp from play store/app store. | Verify that user can set chat wallpaper. |
| 2. | Verify that user is able to register using mobile number on whatsapp. | You can try to send more that 30 images at a time. |
| 3. | Verify that chats window contains the entire chat list with profile picture. | Verify that you can able to send large videos. |
| 4. | Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person. | Try to send hundreds of messages at the same time to a person and then check the behavior of this application. |
| 5. | Verify that user can check the message delivered and read time for a message in the message info section. | Try to register with the same number of two different devices at the same time. |
| 6. | Verify that user can share or receive a contact with the other person. | Check that you are able to forward a message to more than five people. |
| 7. | Verify that user can create a group adding multiple persons from his contact list. | Try to add more than 256 members in a group. |
| 8. | Verify that user can send and receive message in group chats. | Verify whatsapp payment option |
| 9. | Verify that user can send and receive images, audio, video, emoticons in chat to individual. | Verify that user make whatsapp calls to the person who is not in his contact list. |
| 10 | Verify that user can send and receive chats in secondary language. | Verify that user can block a user to prevent any message from getting received from the blocked contact. |
| 11. | Verify that a whatsapp group should have at least 2 or more member. | Verify that user can archive chats in an individual a group chat. |

**Write Scenario of Microwave Oven?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive Test Scenario** | **Negative Test Scenario** |
| 1. | Verify the dimensions of the oven are as per the specification provided. | Check ovens condition when it runs for specific hours. |
| 2. | Verify that the ovens material is optimal for its use as an oven and as per the specification. | Check that disconnecting power while cooking is in progress. |
| 3. | Verify that the oven heats the food at the desired temperature properly. | Verify the battery requirement of the microwave oven and check that its functions smoothly at that power. |
| 4. | Verify that oven heats food at the desired temperature within a specified time duration. | Verify the digital display is clearly visible. |
| 5. | Verify the ovens functioning with maximum attainable temperature. | Check the maximum capacity of the oven and test its functionality. |
| 6. | Verify the ovens functioning with minimum attainable functioning. | Verify the ovens functionality with different kind of container material. |
| 7. | Verify that the ovens plate rotation is speed is optimal and not too high to spill the food kept over it. | Verify that the usage instruction or user manuals have clear instruction. |
| 8. | Verify that the ovens door opens smoothly. | Verify the power cord of the oven is long enough. |
| 9. | Verify that the ovens door get closed properly. | Check maximum capacity of the oven and check its functionality. |
| 10. | Verify that the temperature regulator is smooth to operate. | It should be light in weight. |
| 11. | Check the ovens functionality with different kind of food. | Check if there is no corrosion on the door, the door hinges, or the oven interior. |

**Write scenario of coffee wending machine?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Verify that the dimension of the coffee vending machine is as per the specification. | Check the power button of the coffee wending machine is working correctly after pressing the power button. |
| 2. | Verify the outer body, as well as inner parts material, is as per the specification. | Check the indicator lights are displaying correctly when the coffee vending machine is going to switch off or run. |
| 3. | Verify the machines body color as well as brand is correctly visible as per specification. | Check all the buttons of coffee vending machine have an image and text on them, which indicates what task will be performed if you press the button. |
| 4. | Verify the power voltage requirement of the machine. | Check the complete quantity of coffee poured in a single operation, and no coffee are stored in the nozzle area. |
| 5. | Verify coffee should not leak when not in operation. | Check the coffee vending machine is working as expected. |
| 6. | Verify the amount of coffee served in single-serving is as per specification. | Check the temperature of is served coffee should be the same temperature. |
| 7. | Verify that the digital display displays correct information. | Check if some ingredients if finished. Then it should display an error message on LED screen. |
| 8. | Check if the machine can be switched on or off using the power buttons. | Check the time they take to serve coffee should be the same as mentioned in the specification document. |
| 9. | Verify the mechanism to clean the system work correctly. | Check the functioning of coffee machine when two buttons are pressed simultaneously. |
| 10. | Verify the machine should not make too much sound when in operation. | Check in the coffee, the quantity of water and coffee beans, and the milk mix as expected. |
| 11. | Check there should be a button for passes an extra coffee in the machine. | Check the outer body of the machine looks similar as described in the specification document. |

**Write a scenario of chair?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | To verify that all the parts of the chair are available. | To verify the balance of the chair with one arm. |
| 2. | To verify that all the chair’s legs are fitted properly. | To verify the balance of the chair with three legs. |
| 3. | To verify that the chair cushion is fitted properly or not. | To verify chair stress testing by dropping the chair down from the practical height. |
| 4. | Verify that the both arms are fitted properly or not. | To verify nothing is breaking, no damage to the chair and chair is performed without any issue. |
| 5. | Check that the chair is as per design specification or not. | To verify how the chair is performed in different climate environmental condition. |
| 6. | Check that the chair has the color as per specification or not. | Check the condition when washed with water or effect of water on chair. |
| 7. | Check that the seat as per the specification or not. | Verify if wheels are needed in specified chair. |
| 8. | Check that the logo of the company is printed properly or not. | Check if there is any pointy edge on the chair or not. |
| 9. | Verify the length, breadth, and other size specifications of the chair as per the requirement specification or not. | Check strength and material of the chair. |
| 10. | Check the size and shape should be comfortable for sitting or not. | To verify when putting high pressure on chair for particular amount of time and check the behavior of the chair. |
| 11. | To verify that the material body of the chair. | If the distance between the legs are not proper, it is not usable. |

**Write scenario of face-book chat on mobile?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Check the received message counts should be displayed on face-book message icon. | Check that user can send multiple smiles at a time. |
| 2. | Verify that user gets all received messages in his inbox. | Check that user can send multiple images at the same time. |
| 3. | Verify that only message contacts will display in left hand side of message box. | Check error message displayed or not while share large files. |
| 4. | Verify that profile picture display in left hand side of inbox is correct for each user. | Check that user can able to send messages in local language. |
| 5. | Verify that active user display with green dot in message box. | If internet goes down, user does not get or share messages. |
| 6. | Verify that unread messages are highlighted so that user can identify it. | Check the blocked contact id displaying or not. |
| 7. | Check received messages counts should be displayed with inbox in message page. | Check if the user is not able to send blank message. |
| 8. | Verify that user can search contacts in message box. | Check message functionality working on all supported devices. |
| 9. | Check behavior of chat box if we change network from Wi-Fi to LAN. | Check the scroll bar displays wherever necessary. |
| 10. | Verify that user is able to navigate to old conversion or can view history. | Check user is able to navigate using the tab key. |
| 11. | Verify that message get sent after clicking on enter button. | Check appropriate tooltip given to buttons. |

**Write scenario of Gmail?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Check the recently received unread email is highlighted and bold in the inbox section. | Verify that a different password in confirm password is working fine. |
| 2. | Check if all the elements of the received emails are correctly displayed or not. | Verify that as we hit the next button without entering the user name, then check if it is working. |
| 3. | Check whether the user clicks on the new email; it redirects the user to the email content. | Verify that as we hit the next button without entering the date of birth, then check if it is working. |
| 4. | Verify that any attachments are attached to the email and are downloadable. | Check if all fields are on the page as per the requirement. |
| 5. | Verify that attachments are scanned for viruses before download. | Check is all the compulsory fields are marked or not. |
| 6. | Verify that all email marked as read are not highlighted. | Check for a better user experience checkbox, radio button and dropdown are used on the Gmail registration page. |
| 7. | Verify that count of unread emails is alongside inbox text in the left sidebar of Gmail. | Check the position of the reset, submit and cancel buttons location. |
| 8. | Verify that unread email count increases by one on receiving a new email. | Check after filling all mandatory fields. If the user hits the submit button, then that should be sent to the server. |
| 9. | Verify that email recipients in cc are visible to all users. | Check maximum allowed character for the input field. |
| 10. | Verify that email can be received from non-Gmail id, email ids. | Verified that a newly received email is displaying as highlighted or not. |
| 11. | Verify that email recipient in bcc are not visible to the user. | Verify that a newly received email has correctly displayed sender email id or name, mail subject and mail. |

**Write scenario of online shopping product flip-cart?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | User navigation through all the pages of the application. | If an item is out of stock, a user cannot add it to cart. |
| 2. | None of the links in the application should be visible. | Quantity of symbols allowed. |
| 3. | Company logo, products, prices and their description should be visible. | Specific password requirements. |
| 4. | Product should be listed categories wise on the application. | The account with the given username is not registered in the system it can’t login. |
| 5. | Product should be displayed which match the search criteria. | A user signs in with an old password entered after password change it can’t login. |
| 6. | Relevant product should be listed on the top of the search results page. | A user presses the back button after logging out it remains logged out. |
| 7. | Adding a product to the cart should be possible. | A user signs in with an unverified email address it can’t login. |
| 8. | User should be able to buy the product which is added to the cart once the user signed in. | The navigation in e-commerce site is quite complex. |
| 9. | The entire products which are added to the cart should be purchasable. | It takes time for a user to login with the correct credential. |
| 10. | All the payment methods should be displayed and all of the methods should be working correctly. | A user enters incorrect credentials can’t login |
| 11. | Error message should be displayed on the UI when shipping is not available to the delivery location. | Verify that user should be able to apply coupons or vouchers at checkout from cart. |

**Write a scenario of wrist watch?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Check the watch hour section is working properly or not. | To verify that watch is working without power or power cell or not. |
| 2. | Check that the watch minute section is working properly or not. | To verify that the watch stress testing by dropping the watch down from the desired height. |
| 3. | Check the watch’s second section is working properly or not. | To verify that the watch is working at different climate environmental conditions. |
| 4. | To verify that watch is changing the location of the minute needle as second’s complete 60 seconds. | To verify that watch can use GPS. |
| 5. | To check that date is changing after 24 hours or not if date feature is available in digital watch. | If the dimension of the watch is not proper, it is uncomfortable to wear. |
| 6. | To check that user can change the hours, minute, date, etc. from the wind up function. | If the dial has no glass covering or plastic, it is breakable. |
| 7. | Check the belt of the watch. | If the internal parts are not properly fitted, it’s not working. |
| 8. | Verify the type of watch where it is analog or digital. | Check the design of the watch is as per the specification. |
| 9. | Verify the material of the watch and strap. | To check the length, breadth, and width of the watch |
| 10. | Verify the weight of the watch. | Check the body of the watch can be bent. |
| 11. | Verify the company logo is clearly visible. | To verify the watch is not making sound. |

**Write a scenario of lift?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Verify the dimension. | Wait outside for some-time when the door is opened. |
| 2. | Verify the type of the door is as per the specification. | When inside, observe door closing operations without pressing the closing button. |
| 3. | Verify the type of metal used in the lift interior and exterior. | Check the lift is working when the power is off. |
| 4. | Verify the capacity in terms of the total weight. | Observe the lift working when the power is low voltage. |
| 5. | Verify the buttons in the lift to close and open the door and numbers as per the number of floor. | Verify the door closing operation when the heavy load is in the lift. |
| 6. | Verify that lift move to the particular floor as the button of the floor clicked. | Add many people regardless of weight and check the volume limitations. |
| 7. | Verify that lift stop when up/down buttons on a particular floor is pressed. | Try pressing the open button while lift is moving. |
| 8. | Verify if emergency button to contact officials in case of any mishap. | Press the stop button before reaching the specific floor. |
| 9. | Verify the time duration for which the door remains open by default. | Put small obstacles near the sensors that detect auto-close functionality. |
| 10. | Verify that the lift interior has proper air ventilation. | Create smoke or fire inside the lift. |
| 11. | Verify lighting in the lift. | Add more people than the specified weight. |

**Write scenario of Instagram?**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Positive test scenario** | **Negative test scenario** |
| 1. | Verify that user can download the app from play store/ app store | Check if the stories visible after 24 hours precisely |
| 2. | Verify that user can share photos, videos in chat box. | Check if stranger can check the story in a private account. |
| 3. | Verify the search bar, where we can search other users. | Check if general contact can check the story if it is supposed to be closed list. |
| 4. | Verify to make video-calls. | Check if the blocked/hidden contact can check the story. |
| 5. | Verify that we can watch the reels, and create the reels. | Check in private account, the stranger can send messages. |
| 6. | Verify the like, comment and share button. | Upload a duplicate name and images |
| 7. | Verify that user can chat in secondary languages. | The file section dialog shows only supported files listed. |
| 8. | Verify that user can un-send the messages, audio, video in chat box. | Without internet we cannot use it. |
| 9. | Verify that user can block a user to prevent any messages. | Without internet we cannot create or watch reels. |
| 10. | Check image upload path. | Consuming more battery as well as internet. |
| 11. | Test if user can see the uploaded images. | We cannot create any group. |