**Module–2(Manual Testing)-Que.-Ans.**

1. What is Exploratory Testing?

* Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously.

1. What is traceability matrix?

* To protect against changes you should be able to trace back from every system component to the original requirement that caused its presence.

1. What is Boundary value testing?

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

1. What is Equivalence partitioning testing?

* Equivalence partitioning is the process of defining the optimum number of tests by: Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition within a function.

1. What is Integration testing?

* Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

1. What determines the level of risk?

* A properly designed test that passes, reduces the overall level of Risk in a system.

1. What is Alpha testing?

* It is always performed by the developers at the software development site. Alpha Testing is not open to the market and public, It is always performed in Virtual Environment.

1. What is beta testing?

* It is always performed by the customers at their own site, Beta Testing is always open to the market and public. It is performed in Real Time Environment.

1. What is component testing?

* Component(Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.”

1. What is functional system testing?

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

1. What is 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.

1. What is GUI Testing?

* Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

What is Adhoc testing?

* Adhoc testing is an informal testing type with an aim to break the system.

1. What is load 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.

1. What is stress Testing?

* 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.

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

* White Box Testing: Testing based on an analysis of the internal structure of the component or system.

Types of white box testing:- 1)Statement coverage,2)Branch coverage,3)Decision coverage.

1. What is black box testing? What are the different black box testing techniques?

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

Black box testing techniques :-1) Equivalence partitioning, 2) Boundary value analysis, 3) Decision tables ,4) State transition testing, 5) Use-case Testing, 6) Other Black Box Testing, 7) Syntax or Pattern Testing.

1. Mention what are the categories of defects?

* Defect Category: Security, Database, Functionality (Critical/General), UI

Mention what bigbang testing is?

* simultaneously In Big Bang integration testing all components or modules is integrated, after which everything is tested as a whole.

What is the purpose of exit criteria?

* (1) Successful Testing of Integrated Application.(2) Executed Test Cases are documented.(3) All High prioritized bugs fixed and closed.(4) Technical documents to be submitted followed by release Notes.

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

🡪 This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

What is 7 key principles? Explain in detail?

* 1.Testing shows presence of Defects 2. Exhaustive Testing is Impossible! 3. Early Testing 4. Defect Clustering 5. The Pesticide Paradox 6. Testing is Context Dependent 7. Absence of Errors Fallacy
* **Testing shows presence of Defects**:-Testing can show that defects are present, but cannot prove that there are no defects. Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness. We test to find Faults As we find more defects, the probability of undiscovered defects remaining in a system reduces. However Testing cannot prove that there are no defects present.
* **Exhaustive Testing is Impossible**:-Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts. For example: In an application in one screen there are 15 input f ields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests. This is very unlikely that the project timescales would allow for this number of tests. So, accessing and managing risk is one of the most important activities and reason for testing in any project. We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions). That is we must Prioritise our testing effort using a Risk Based Approach.
* **Early Testing** :-Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. Testing activities should start as early as possible in the development life cycle These activities should be focused on defined objectives – outlined in the Test Strategy Remember from our Definition of Testing, that Testing doesn’t start once the code has been written!
* **Defect Clustering** :-A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures. defects are not evenly spread in a system They are ‘clustered’ 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 An important consideration in test prioritisation!
* **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 As bugs are eliminated by the programmers, the software improves As software improves the effectiveness of previous tests erodes. Therefore we must learn, create and use new tests based on new techniques to catch new bugs N.B It's called the "pesticide paradox" after the agricultural phenomenon, where bugs such as the boll weevil build up tolerance to pesticides, leaving you with the choice of ever-more powerful pesticides followed by ever-more powerful bugs or an altogether different approach.’ – Beizer 1995.
* **Testing is Context Dependent**:- Testing is basically context dependent. Testing is done differently in different contexts. Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing 3 to 10 failures per thousand lines of code (KLOC) typical for commercial software 1 to 3 failures per KLOC typical for industrial software 0.01 failures per KLOC for NASA Shuttle code! Also different industries impose different testing standards Safety – critical software is tested differently from an e-commerce site. Different kinds of sites are tested differently. Safety – critical software is tested differently from an e-commerce site.
* **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. If we build a system and, in doing so, find and fix defects .... it doesn’t make it a good system Even after defects have been resolved it may still be unusable and/or does not fulfil the users’ needs and expectations.

Difference between QA v/s QC v/s Tester

|  |  |  |  |
| --- | --- | --- | --- |
| SR.NO. | Quality Assurance | Quality Control | Testing |
|  | Activities which ensure the implementation of 1 2 3 4 5 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. |
|  | 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. |
|  | Process oriented activities. | Product oriented activities. | Product oriented activities. |
|  | Preventive activities. | It is a corrective process. | It is a preventive process. |
|  | 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. |

Difference between Smoke and Sanity?

|  |  |  |
| --- | --- | --- |
| Sr. No. | **Smoke** | **Sanity** |
|  | 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 |
|  | 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. |
|  | This testing is performed by the developers or testers | Sanity testing is usually performed by testers |
|  | Smoke testing is usually documented or scripted | Sanity testing is usually not documented and is unscripted |
|  | Smoke testing is a subset of Regression testing | Sanity testing is a subset of Acceptance testing |

Difference between verification and Validation

|  |  |  |
| --- | --- | --- |
| Criteria | **Verification** | **Validation** |
| Definition | The process of evaluating work-products (not the actual f inal 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. |
| 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 Plans, Requirement Specs, Design Specs, Code, Test Cases ∙ Reviews ∙ Walkthroughs when placed in its intended environment. |
| Question | Question Are we building the product right? | Are we building the right product? |
| Evaluation Items | Plans, Requirement Specs, Design Specs, Code, Test Cases | The actual product/software. |
| Activities | ∙ Reviews ∙ Walkthroughs  ∙ Inspections | ∙ Testing |

Explain types of Performance testing.

* Types of Performance Testing:- (1) Load testing (2) Stress testing (3) Endurance testing (4) Spike testing (5) Volume testing (6) Scalability testing.

27. Explain the difference between Functional testing and Non Functional testing

|  |  |
| --- | --- |
| **Functional testing** | **Non functional testing** |
| Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements. | Non functional testing checks the performance,reliability,scalability and other non functional aspects of the software system. |
| Functional testing is executed first | Non functional testing should be performed after functional testing |
| Manual testing or automation tools can be used for functional testing | Using tools will be effective for this testing |
| Business requirements are the inputs to functional testing | Performance parameters like speed , scalability are inputs to non-functional testing. |
| Functional testing describes what the product does | Nonfunctional testing describes how good the product works |
| Easy to do manual testing | Tough to do manual testing |
| 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 |

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

|  |  |
| --- | --- |
| STLC | SDLC |
| Entry Criteria: Entry Criteria gives the prerequisite items that must be completed before testing can begin. | SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are a number of different development models. |
| Exit Criteria: Exit Criteria defines the items that must be completed before testing can be concluded | A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. |
| 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 :- Requirements Collection/Gathering ,Analysis ,Design ,Implementation ,Testing, Maintenance |

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

🡪 **Test Scenario**:- A Scenario is any functionality that can be tested. It is also called Test Condition, or Test Possibility. **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. **Test Case:-** Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks. That is the difference between test scenarios,test cases and test script.

30. 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, Determining the scope and risks, and identifying the objectives of testing. Defining the amount, level of detail, structure and templates for the test documentation.

31. What are the different Methodologies in Agile Development Model?

🡪 The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating. 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 (say- 7 to 9 members). It consists of three roles and their responsibilities are explained as follows: Scrum Master: Master is responsible for setting up the team, sprint meeting and removes obstacles to progress Product owner: The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration Scrum Team: Team manages its own work and organizes the work to complete the sprint or cycle Sprint: Sprint is a time-boxed period in which the scrum team needs to finish the set amount of work. Each sprint has a specified timeline, i.e., 2 weeks to 1 month. The scrum team agrees with this timeline during the sprint planning meeting. **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. Kanban originated in Toyota in the 1940s. Kanban’s meaning in Japanese is “billboards.” The Kanban board has columns and story cards. The columns are nothing, but workflow states and cards are nothing but a demonstration of the actual task a team member is performing.

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

🡪 Authorization **:-** Authorization is the process of determining the level of access a user has within an application, based on their role or permissions.

Once a user is authenticated, the authorization system controls which resources they can access and what actions they can perform.

**Authentication :-** Authentication refers to the process of verifying a user’s identity, usually by requiring them to provide valid credentials (e.g., username and password).

The primary goal of authentication is to confirm that a user is who they claim to be before granting access to the application.

Differences between Authentication and Authorization:

* + While authentication focuses on verifying a user’s identity, authorization is concerned with managing their access to specific resources and actions.
  + Authentication typically occurs at the beginning of a user session, while authorization is an ongoing process that may be evaluated multiple times during a session.
  + Implementing both authentication and authorization is essential to protect sensitive data and maintain the integrity of your web applications.

33. Write a scenario of only Whatsapp chat messages.

* 🡪 Verify that users can send text messages to individual contacts.
* Confirm that sent messages are immediately delivered.
* Test sharing images, videos, and documents in individual chats.
* Verify that users can use emojis and stickers in individual chats.
* Test sending and receiving voice messages in individual chats.
* Verify that users can share contacts and locations in individual chats.
* Verify that messages display accurate timestamps.
* Confirm that read receipts (blue ticks) are displayed accurately unless stated otherwise in the privacy setting of the receiver.
* Verify that users can react to individual chat messages.
* Verify that users can mute notifications for a specific individual chat.
* Test deleting individual chat messages.

34. Write a Scenario of Pen.

1. Verify that the length and the diameter of the pen are as per the specifications.
2. Verify the outer body material of the pen. Check if it is metallic, plastic, or any other material specified in the requirement specifications.
3. Check the color of the outer body of the pen. It should be as per the specifications.
4. Verify that the brand name and/or logo of the company creating the pen should be clearly visible.
5. Verify that any information displayed on the pen should be legible and clearly visible.
6. Check the odor of the pen’s ink on writing over a surface.
7. Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc.
8. Verify that the text written by the pen should have consistent ink flow without leaving any blob.
9. Check that the pen’s ink should not leak in case it is tilted upside down.
10. Verify if the pen’s ink should not leak at higher altitudes.
11. Verify if the text written by the pen is erasable or not.
12. Verify the functioning of a pen at extreme temperatures – much higher and lower than room temperature.
13. Verify the functioning of a pen at extreme altitude.
14. Check the functioning of a pen at zero gravity.
15. Verify the functioning of the pen by applying extreme pressure.
16. Verify the effect of oil and other liquids on the text written with a pen.

35. Write a Scenario of Pen Stand.

🡪 1. Verify that the how big/small size of the pen stand are.

2. Verify that which type of material are used in making the pen stand (ex. Plastic, metal, wood etc.)

3. Verify the stand height is enough for put the pen inside it.

4. Verify the weight of the pen stand not more heavy not less because in stand we can put one or more pens inside it.

5. Verify that the pen stand is with clock/calendar or not.

6. Verify that pen stand with proper grip of bottom of the stand for easily put it on any surface like tile, table, glass etc.

7. Verify that outer body of pen stand has latest colour and design so customer attract to purchase it.

36. Write a Scenario of Door.

1. 🡪 Verify if the door is single door or bi-folded door.
2. Check if the door opens inwards or outwards.
3. Verify that the dimension of the doors are as per the specifications.
4. Verify that the material used in the door body and its parts is as per the specifications.
5. Verify that color of the door is as specified.
6. Verify if the door is sliding door or rotating door.
7. Check the position, quality and strength of hinges.
8. Check the type of locks in the door.
9. Check the number of locks in the door interior side or exterior side.
10. Verify if the door is having peek-hole or not.
11. Verify if the door is having stopper or not.
12. Verify if the door closes automatically or not – spring mechanism.
13. Verify if the door makes noise when opened or closed.
14. Check the door condition when used extensively with water.

37. Write a Scenario of ATM

1. 🡪 Verify that all the labels and controls including text boxes, buttons, images, and links are present on the screen.
2. Check the informative text written displayed on the screen is clearly visible and legible.
3. Verify that the size, color, and UI of the different objects are as per the specifications.
4. Verify that the application’s UI is responsive i.e. it should adjust to different screen resolutions of ATM machines.
5. Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both.
6. Verify that on properly inserting a valid card different banking options appear on the screen.
7. Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly.
8. Verify that the touch of the ATM screen is smooth and operational.
9. Verify that the user is presented with the option to choose a language for further operations.
10. Check that the user is asked to enter a pin number before displaying any card/bank account detail.
11. Verify that there is a limited number of attempts up to which the user is allowed to enter the pin code.
12. Verify that if the total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further. And operations like temporary blocking of the card, etc get initiated.
13. Check that the pin is displayed in masked form when entered.
14. Verify that the user is presented with different account type options like- saving, current, etc.
15. Verify that the user is allowed to get account details like available balance.
16. Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal.
17. Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications.
18. Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured.
19. Check that the user cannot withdraw more amount than the total available balance and a proper message should be displayed.
20. Verify that the user is provided the option to get the transaction details in printed form.
21. Verify that the user’s session timeout is maintained.

38. When to used Usablity Testing?

🡪 **Need For 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, due to following reasons 1.Where do I click next? 2.Which page needs to be navigated? 3.Which Icon or Jargon represents what? 4.Error messages are not consistent or effectively displayed 5.Session time not sufficient. Usability Testing identifies usability errors in the development cycle and can save a product from failure.

39. What is the procedure for GUI Testing?

🡪 Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

40. Write a scenario of Microwave Owen

1. Verify that the dimensions of the oven are as per the specification provided.
2. Verify that the oven’s material is optimal for its use as an oven and as per the specification.
3. Verify that the oven heats the food at the desired temperature properly.
4. Verify that the oven heats food at the desired temperature within a specified time duration.
5. Verify the ovens functioning with the maximum attainable temperature.
6. Verify the ovens functioning with minimum attainable temperature.
7. Verify that the oven’s plate rotation speed is optimal and not too high to spill the food kept over it.
8. Verify that the oven’s door gets closed properly.
9. Verify that the oven’s door opens smoothly.
10. Verify the battery requirement of the microwave oven and check that it function’s smoothly at that power.
11. Verify that the text written over the oven’s body is clearly readable.
12. Verify that the digital display is clearly visible and functions correctly.
13. Verify that the temperature regulator is smooth to operate.
14. Verify that the temperature regulator works correctly.
15. Check the maximum capacity of the oven and test its functioning with that volume of food.
16. Check the oven’s functionality with different kinds of food – solid, and liquid.

41. Write a scenario of Coffee vending Machine

1. Verify that the dimension of the coffee machine is as per the specification.
2. Verify that outer body, as well as inner part’s material, is as per the specification.
3. Verify that the machine’s body colour as well brand is correctly visible and as per specification.
4. Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc.
5. Verify that the quantity of hot water, milk, coffee powder per serving is correct.
6. Verify the power/voltage requirements of the machine.
7. Verify the effect of suddenly switching off the machine or cutting the power. The machine should stop in that situation and in power resumption, the remaining coffee should not get come out of the nozzle.
8. Verify that coffee should not leak when not in operation.
9. Verify the amount of coffee served in single-serving is as per specification.
10. Check if the machine can be switched on and off using the power buttons.
11. Check for the indicator lights when the machine is switched on-off.
12. Verify that the functioning of all the buttons work properly when pressed.
13. Verify that each button has an image/text with it, indicating the task it performs.
14. Check the performance of the machine when used continuously until the ingredients run out of the requirements.
15. Check the functioning of coffee machine with a lesser or higher voltage than required.
16. Check the functioning of the coffee machine if the ingredient container’s capacity is exceeded.

42. Write a scenario of chair.

1. 🡪 Verify that the chair is stable enough to take an average human load.
2. Check the material used in making the chair-wood, plastic etc.
3. Check if the chair’s leg are level to the floor.
4. Check the usability of the chair as an office chair, normal household chair.
5. Check if there is back support in the chair.
6. Check if there is support for hands in the chair
7. Verify the paint’s type and color.
8. Verify if the chair’s material is brittle or not.
9. Check if cushion is provided with chair or not.
10. Check the condition when washed with water or effect of water on chair.
11. Verify that the dimension of chair is as per the specifications.
12. Verify that the weight of the chair is as per the specifications.

43. Write a Scenario of Wrist Watch.

1. 🡪 Verify the type of watch – analog or digital.
2. In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch.
3. In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed.
4. Verify the material of the watch and its strap.
5. Check if the shape of the dial is as per specification.
6. Verify the dimension of the watch is as per the specification.
7. Verify the weight of the watch.
8. Check if the watch is waterproof or not.
9. Verify that the numbers in the dial are clearly visible or not.
10. Check if the watch is having a date and day display or not.
11. Verify the color of the text displayed in the watch – time, day, date, and other information.
12. Verify that clock’s time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.
13. Check if the second hand of the watch makes ticking sound or not.
14. Verify if the brand of the watch and check if its visible in the dial.
15. Check if the clock is having stopwatch, timers, and alarm functionality or not.
16. In the case of a digital watch, verify the format of the watch 12 hours or 24 hours.
17. Verify if the watch comes with any guarantee or warranty.
18. Verify if the dial has glass covering or plastic, check if the material is breakable or not.

44. Write a Scenario of Lift (Elevator)

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