**TOPS TECHNOLOGIES AHMEDABAD**

**MODULE 2**

**1 What is Exploratory testing ?**

**Answer :-** A Exploratory testing is a software testing approach where testers simultaneously design and execute tests to learn about the software's behavior and identify potential issues.

Unlike traditional testing with predefined test cases, exploratory testing emphasizes flexibility, where the tester's understanding of the software evolves as they test, allowing for dynamic adaptation and discovery of unexpected bugs.

**2) What is traceability matrix ?**

**Answer :- RTM ( Requirement Traceability matrix )**

A Requirement Traceability matrix is a document that maps and traces Software requirement to test cases , ensuring that all requirements are tested and validated.

* It helps verify that the software its intended functionality and aids in identifying potential gaps or missing requirement
* RTM provides clear path for testing , change impact analysis and project and tracking .

**3) what is boundary value testing ?**

**Answer :-** Tests values at the boundary of input ranges .

* Example:- For a field accepting 1-100
* Test 0,1,2,99,100,101,

**4 ) What is Equivalence partioning testing ?**

**Answer:-** Divides input data into valid partitions

* Test case are designed to cover each parition .
* Example:- A login field accepting ages 18-60.

Valid parition 18-60

Invalid parition <18>60 .

**5) what is a integration testing ?**

**Answer:-** Integration Testing in software testing means testing how different modules or components of a software system work together.

When developers build software, they usually make it in small parts (modules).

Each module may work fine on its own (tested in Unit Testing).

But when you connect these modules, there can be issues in data flow, communication, or interaction.

Integration Testing checks if modules interact correctly and the system works as expected when combined.

**Example:**

Suppose you have a food delivery app:

Module 1: Login

Module 2: Search restaurant

Module 3: Add to cart

Module 4: Payment

**6) what determines the level of risk ?**

**Answer:-**  A factor that could result in future negative consequences usually expressed as impact and likelihood.

Need to identify the risks associated with the your project

Types of Risks

* product risks - functionality risks , reliability risks , security risks , performance risks .
* Project risks - schedule risks , budget risks , resource risks , communication risks .
* Technical risks
* Tool risks
* Requirements risks

**7) what is alpha testing ?**

**Answer:-** Alpha testing is an internal test .

**8) what is beta testing ?**

**Answer:-** Beta testing is an external pilot test before the official release.

**9) what is component testing ?**

**Answer:-** It is a type of software testing where we test one small part (component/module) of the software separately to check if it works correctly.

**Example:**

In a calculator app, testing only the addition function (without checking subtraction, multiplication, or the whole app) is component testing.

In a shopping app, testing only the login module is component testing.

**10) what is functional system testing ?**

**Answer:-** Functional system testing in software testing means checking the entire software system as a whole to ensure it works according to the specified functional requirements.

It is testing the complete application’s functions (end-to-end) to see if the system behaves as expected when all components/modules are integrated together.

**Example:**

For an E-commerce application:

Add product to cart

Apply discount code

Make payment through UPI

Get order confirmation

**11) what is non functional system testing ?**

**Answer :-** Testing the attribute of a component or system that do not relate to functionality , e.g. Reliability , efficiency , usability , interoperability , maintainability and portability

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

E.X = web based testing , desktop testing , mobile based testing , game based testing .

**12 ) What is GUI testing ?**

**Answ**er  **A Graphical user interface ( GUI )**

* Focus on validating the visual elements and interactions of software applications
* Primary goal of GUI testing is to ensure that the application user interface functions correctly.
* **Key aspects**

Visual elements validation

Error handling

Functionality and cross browser consistency

User experience (UX) usability

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

**Answer :-** Adhoc testing in software testing refers to an informal and unstructured testing approach where testers explore a software application without following any predefined test plans, test cases, or documentation. It relies heavily on the tester's intuition, experience, and understanding of the software to uncover defects that might be missed by formal, structured testing methods.

Adhoc testing is particularly useful when time for testing is limited, or when a quick assessment of a prototype or new feature is required. It complements formal testing by providing a different perspective and can be highly effective in identifying critical issues early in the development cycle.

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

**Answer :-**  Load testing in software testing is a type of performance testing that evaluates the behavior of a software application or system under a simulated, expected user load. Its primary goal is to determine if the system can handle the anticipated number of users and transactions while maintaining acceptable performance levels, such as response times and stability.

**Q.15) what is Stress testing ?**

**Answer :-** Stress testing in software testing is a non-functional testing technique that evaluates the robustness and stability of a software system or application under extreme and abnormal load conditions. The primary goal is to determine the system's breaking point, identify how it behaves under stress, and assess its ability to recover gracefully from failure.

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

**Answer:-** White Box Testing in software testing is a method where the tester looks inside the program’s code, logic, and structure to design test cases.

The tester checks how the code works internally instead of just checking the output.

1. **Statement Coverage**
2. **Branch / Decision Coverage**
3. **Path Coverage**
4. **Condition Coverage**

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

**Answer:-** These techniques focus on testing the functionality of the software without looking at its internal code structure.

**1. Equivalence Partitioning (EP)**

Divides input data into different groups (partitions) where test cases can be considered equivalent.

**Example: If age must be between 18–60, then partitions are**:

Below 18 (Invalid)

Between 18–60 (Valid)

Above 60 (Invalid)

**2. Boundary Value Analysis (BVA)**

Tests the boundaries of input values (just below, on, and just above the boundary).

**Example: For age between 18–60:**

Test with 17, 18, 60, 61

**3. Decision Table Testing**

Useful when there are different combinations of inputs with different outcomes.

Represents input conditions and their actions in a table form.

**Example: Login feature with conditions:**

Correct Username + Correct Password → Success

Correct Username + Wrong Password → Error

Wrong Username + Correct Password → Error

**4. State Transition Testing**

Tests the behavior of an application for different states based on input events.

**Example:**

ATM machine:

Insert card → Enter PIN → Withdraw cash

Wrong PIN 3 times → Card blocked

**5. Use Case Testing**

Based on real-world scenarios and user workflows.

**Example: In an e-commerce app:**

User adds item to cart → Proceeds to checkout → Makes payment → Order confirmation.

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

**A 1. Severity :**

Defects that causes system failure ,server functionality loss ,or safety hazards , requiring immediate attention.

Defects that significantly affect functionality but don’t cause system failure, impacting usability or user experience.

Defect that have a limited impact on functionality or usability, like cosmetic issues or minor inconvenience.

1. **Priority :**

Defect that must be fixed before any other, preventing release or impacting critical operations

Defect that need to be fixed quickly often before other high priority issues.

Defects that should be addressed as soon as possible impacting functionality

Defects that can be addressed after high – priority issues are resolved.

Defects that can be deferred and fixed in later releases.

1. Type (in software):

Functional: Defects related to the core functionality of the software not working as specified in the requirements.

Performance: Defects related to speed, deficiency, or resource usage of the software.

Usability: Defects related to the user experience, such as ease of use, navigation, or interface design.

Security :

documentation.

1. **Other Categorizations:**

Flaws in the product's design that make it inherently dangerous or inactive.

Defects that occur during the manufacturing process.

Inadequate warnings or instructions accompanying the product.

**Q.19) Mention what big bang testing is?**

**Answer :-** The big bang approach in software testing is a method where all components of a system are integrated together at once and then tested as a whole. This means individual modules are not integrated or tested separately beforehand; instead, they are combined into a single unit, and the entire system is tested together.

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

**Answer:-** Purpose of exit criteria is to define when we stop testing either at the :

End off all testing - i.e. product go live

End of phase testing

Exit criteria is to clearly define the condition that must be met before

a testing phase or any other process can be consider complete.

Exit criteria are the criteria or requirements which must be met to complete a specific task or process as used in some fields of business or science ,such as software engineering .

All test cases have been executed.

A specific – priority defects have been resolved .

The software meets performance criteria.

Stakeholders have signed off on the software.

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

**Answer :- Regression testing**

This is a crucial part of maintenance testing , involving , the re execution of existing test cases to ensure that the recent modifications have not inadvertently introduce new bugs or broken previously working functionality within other part of the system.

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

**Answer :- 7 key principles are as under**

1. **Testing Shows the presence of defects.**
2. **Exhausting Testing is impossible.**
3. **Early Testing .**
4. **Defect clustering .**
5. **The pesticide paradox.**
6. **Testing is context dependent.**
7. **Absence of error fallacy.**

**(1) Testing shows presence of defects :-**

Testing can shows that defects are present but can not prove that there are no defects.

Testing reduces the probability of undiscovered deflects remaining in the software but even if no defects are found it is not proof of correctness

We test to find result .

**(2) Exhausting Testing is impossible :-**

Testing everything including all combinations of inputs and precondition is not possible

* So instead of doing the exhausting we can use risks and priorities to focus testing efforts.
* Accessing and managing risk is one of the most important activities and reasons for testing in any project .
* That is we must priorities our testing effort using a risk base approach.

**(3) Early Testing:-**

Testing Activity should start as early as possible in the SDLC and should be focused on defined objectives.

Testing activity should be focused on defend objectives outline in the test strategy.

(**4) Defect clustering :-**

A small number of modules contain most of all the defect discovered during prerelease testing or are responsible for the must operational failures.

* Defects are not evenly spread in a system they are clustered.
* Most defects found during testing are usually confined to small number of module.

(**5) Pesticide paradox :-**

If the same tests are repeated over and over again the same set of test cases will no longer find any new defects.

* The tests cases need to be regularly reviewed and revised and new and different need to be written to exercise different part of the software or system to potentially find more defects.
* There for we must learn , create and use new test base on new techniques to catch new bugs.
* A bugs are the eliminated by the programmer the software improves the effectiveness of previous test erodes.

(**6) Testing is context dependent:-**

Testing is context dependent, testing is done differently in different contexts.

* Ex . safety - critical software is tested differently from an e commerce site.
* 3 to 10 failures per thousand line of code typically for commercial software
* 1 to 3 failure per kloc typically for industries software
* 0.01 failure per kloc Nasa shuttle code.
* Different industries impose different testing standards.

**(7) Abuse of errors fallacy :-**

If the system built is unusable and does not fulfill then users , users need and expectations then finding and fixing defects does not help .

If we build a system and in doing so find and fix defects

Evern after resolve it may still be and /or .

**Q.23) Difference between Q/A and Q /C tester ?**

**Answer:- QA ( Quality Assurance )**

QA focuses on providing assurance that the quality requested will be achieved .

QA is the technique of managing quality

QA involved during the development phase

QA does not include the execution of the program

QA is managerial tool

QA is process oriental

QA is a preventive technique

QA is a proactive measure.

QA is responsible for the entire software development life cycle (SDLC)

QA is low level activity that identify an error and mistake that QC cannot .

* **QC ( Quality Control )**

QC focuses on fulfilling the quality requested

QC is the technique to verify quality

QC is not included during the development phase

QC always include the execution of the program

QC is a corrective tool

QC is product oriental

QC is performed after QA activity

QC is a corrective technique

QC is a reactive measure

QC is responsible for the software testing life cycle (STLC )

QC is high level activity that identify on error that QA cannot

QC is more time consuming activity .

**Q.24) Difference between smoke and sanity ?**

**A Difference in short table are as under**

**Feature Smoke testing Sanity testing**

Scope Wide but shallow Narrow but deep

Purpose Check if build is stable Verify specific fix or

Feature

When After new build After new change or fix

Coverage All major modules Specific module

Time taken Short Short to medium

**Q.25) Difference between Verification and validation ?**

**Answer:-**

**Verification:-** verification refers to the seat of the activated that ensure software correctly implement the specific function.

Check the software correctly according to specifications.

Verification means are build the product right .

Verification is the static testing .

It does not include the expecation of the card

It does not include modeling of the code.

It can find the bugs in theory

**Validation :-** Validation refer to the set of activities that ensure that the software that has been built is traceable to customer requirements.

Ensure the software meets user needs and expectation.

Verification is Dynamic testing.

It include execution of the code.

It can only find the bugs that could not be found by the verification process.

**Q.26) Explain types of performance testing ?**

**Answer :-** Performance testing is a type of software testing that checks how well an application performs under expected workloads. It helps ensure that the system is stable, fast, and scalable.

Here are the main types of performance testing:

1. **Load Testing**

**What it is:** Tests how the system behaves under expected user loads.

**Goal:** To check response times, throughput, and resource usage under normal and peak conditions.

**Example**: Checking if an e-commerce site can handle 10,000 users browsing at the same time.

**Stress Testing**

**What it is:** Puts the system under extreme workloads until it breaks.

**Goal**: To identify the system’s breaking point and understand how it fails.

**Example**: Continuously increasing users until the website crashes.

1. **Spike Testing**

**What it is:** Tests the system’s reaction to sudden and extreme increases or decreases in load.

**Goal**: To check stability during unexpected traffic spikes.

**Example**: Ticket booking sites during a big concert announcement.

1. **Endurance (Soak) Testing**

**What it is:** Tests the system under a significant load for a long period.

**Goal**: To detect memory leaks, slowdowns, or performance degradation over time.

**Example**: Running a banking application for 24 hours with 1,000 active users.

1. **Scalability Testing**

**What it is:** Checks how well the system scales up or down when more users, data, or resources are added.

**Goal**: To determine maximum capacity and how performance changes with scaling.

**Example**: Adding more servers and checking if response time improves.

1. **Volume Testing**

**What it is:** Focuses on system performance when handling a large amount of data.

**Goal**: To ensure the system can manage high volumes of data without slowing down.

**Example**: Uploading millions of records into a database and checking query response time.

**Load Testing → Normal workload**

**Stress Testing → Beyond limits**

**Spike Testing → Sudden traffic surges**

**Endurance Testing → Long-duration usage**

**Scalability Testing → Growth handling**

**Volume Testing → Large data handling**

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

**1) Error :-** A human action that produce an incorrect result can manifest

**2) Bug :-**  a fault in a program which cause the program is an unintended or unanticipated manner EX. Anomaly, defect , exception and fault .

**3) Defect :-** Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

**4)Failure :-**  Deviation of the component or system from its expected delivery, service or result.

**Q.28) Different between priority and severity ?**

**Answer:-**

| **Aspect** | **Severity (Impact)** | **Priority (Urgency)** |
| --- | --- | --- |
| **Definition** | How badly it affects the system | How quickly it should be fixed |
| **Decided by** | Tester | Project Manager / Client |
| **Focus** | Technical functionality | Business needs |
| **example** | App crashes = High Severity | Wrong logo on homepage = High Priority |

**Q 29) What id Bug lifecycle ?**

**Answer:-** The bug life cycle, also known as the defect life cycle, is the process that a software bug goes through from its initial identification to its final resolution.

Here's a breakdown of the common stages in a bug's life cycle. The bug is initially identified and reported. The bug is assigned to a developer for analysis and fixing.

The developer has started working on the bug. The developer has implemented a fix for the bug. The bug fix is ready for testing, and the tester is waiting to verify it.

The tester verifies the fix to ensure it has resolved the issues.: The tester confirms the fix works as expected. The bug is officially closed, as it has been successfully resolved

**Q.30) Explain the difference between functional testing and non functional testing ?**

**Answer:-**

**Functional testing :-**

It verifies the operations and actions of an application. It is based on requirements of customer. It helps to enhance the behavior of the application.

Functional testing is easy to execute manually. It tests what the product does.

Functional testing is based on the business requirement

**Non Functional testing : -**

It verifies the behavior of an application. It is based on expectations of customer. It helps to improve the performance of the application.

It is hard to execute non-functional testing manually. It describes how the product does.

Non-functional testing is based on the performance requirement.

**Q.30 HLR only first page Instagram.**

**Introduction**

The first page of Instagram allows existing users to log in and new users to sign up.

**Scope** - Provide login for existing users , allow signup for new users , Third-party login (Facebook/Google) Forgot password recovery Show Instagram branding

**Functional Req.** 1. Login: Username/Email/Phone + Password 2. Show/Hide password option 3. Forgot Password link 4. Signup: Register with Email/Phone, Name, Username, Password 5. Social login (Facebook/Google)

**UI/UX Req.** - Instagram logo on top ,Clear buttons: "Log In" & "Sign Up" ,Error messages for invalid input

Security Req. - Encrypted password storage , Secure HTTPS login , Temporary account lock after multiple failed attempts

**Non-Functional Req.** Performance: Load within 2–3 seconds

**Usability**: Simple & intuitive design **Compatibility**: Works on iOS, Android & Web.

**Constraints** Internet connection required. User must be 13+ years to register.

**👉 HLR of face book only first page**

**Introduction :-** The first page of Facebook should allow existing users to securely log in and new users to create an account.

**Scope:-** Provide login for existing users Allow signup for new users password recovery option , Multi-language support Show Facebook branding

**Functional Req :-**  1. Login: Email/Phone Password 2. Show/Hide password option 3. Forgot Password link 4. Signup: Enter First Name, Last Name, Mobile/Email, Password, Date of Birth, Gender 5. Language selection dropdown

**UI/UX Req :-** Facebook logo on top Prominent buttons "Log In" & "Create New Account" Simple form layout for signup Error messages for invalid input

**Security Req** **:-**  Secure HTTPS connection Encrypted password storage. Captcha or verification for suspicious login attempts , Temporary lockout after multiple failed login attempts

**Non-Functional Req :-** Performance: Login/Signup should load within 2–3 seconds

**Usability:-** User-friendly interface **Compatibility:-** Works across web, iOS, and Android platforms

**Scalability:-** Should support large number of concurrent users

Constraints - Internet connection required User must be 13+ years old to register.

**Test Cases**

**Instagram only first page Test cases**

**TC-01 :-**

**Scenario :- verify login the valid username and**

**Password.**

**Test step :- Enter the correct user name and**

**Password to click the login .**

**Expected result:- user can shown the home**

**Page .**

**TC-02 :-**

**Scenario :- Verify login with invalid username**

**Test step :- Enter invalid username + valid password**

**Expected result :- "The username can not valid.**

**TC\_03**

**Scenario :- Verify login with invalid password Test step :- Enter valid username + wrong password**

**Expected result:- Sorry, your password was incorrect”**

**TC\_04**

**Scenario:- Verify blank fields**

**Test Step :- Leave both username & password empty → click login**

**Expected result :-Enter your username/email and password”**

**TC\_05**

**Scenario:- Verify “Forgot password?” link**

**Test step :- Click on “Forgot password?” Expected result :- Redirects to password recovery screen**

**TC\_06**

**Scenario :- Verify login with phone number**

**Test step:-Enter registered phone number + correct password**

**Expected result:- Login successful**

**TC\_07**

**Scenario:- Verify login with email**

**Test Step :- Enter registered email + correct password**

**Expected result :- Login successful**

**TC\_08**

**Scenario :- Verify password masking**

**Test Step :- Enter password in field**

**Expected result :- Password should be masked (dots/bullets)**

**TC\_09**

**Scenario:-Verify “Show password” toggle**

**Test step :- Enable “Show password”**

**Expected result:- Password should be visible**

**TC\_10**

**Scenario:- Verify login button Disable**

**Test Step :- Keep fields empty**

**Expected result :- Login button should remain disabled**

**TC\_11**

**Scenario :- Verify “Continue with Facebook” Test step :- Click on button**

**Expected result :- Redirects to Facebook authentication**

**TC\_12**

**Scenario:- Verify “Sign up” link**

**Test Step :- Click “Sign up”**

**Expected result:- Redirects to sign-up page**

**TC\_13**

**Scenario:- Verify new account creation with valid details**

**Test step :- Enter full name, username, email/phone, password**

**Expected result:- Account should be created, OTP sent for verification**

**TC\_14**

**Scenario:- Verify sign-up with already registered email/phone**

**Test Step :- Enter registered email/phone Expected Result:- Error: “Email already in use”**

**TC\_15**

**Scenario :- Verify OTP verification during sign-up Enter**

**Test step:- valid OTP**

**Expected result:-Account should be verified and login successful.**

**TC\_16**

**Scenario:-Verify invalid OTP**

**Test Step :- Enter wrong OTP**

**Expected result :- Error: “Invalid code, try again”**

**TC\_17**

**scenario :- Verify terms & privacy links**

**Test step:- Click links at bottom**

**Expected result:- Should open respective policy pages**

**TC\_18**

**Scenario:- Verify UI responsiveness**

**Test step :- Open login page on mobile, tablet, desktop**

**Expected result:- Page should be responsive and aligned properly**

**TC\_19**

**Scenario :- Verify security after failed attempts Test Step :- Enter wrong password 5 times Expected Result:- Captcha or temporary lock should trigger**

**TC\_20**

**Scenario:- Verify remember me option**

**Test step :- Login and check “Remember me” Expected Result:- User should stay logged in even after app restart.**

**Q.31 What is the difference between the STLC(Software testing life cycle) and SDLC(Software development life cycle) ?**

**Answer :-** SDLC(Software development life cycle):

• SDLC is mainly related to software development.

• Besides development other phases like testing is also included.

• SDLC involves total six phases or steps.

• in SDLC, more number of members (developers) are required for the whole process.

• In SDLC, development team makes the plans and designs based on the requirements.

• Goal of SDLC is to complete successful development of software.

• SDLC phases are completed before the STLC phases.

• Post deployment support , enhancement , and update are to be included if necessary.

• Creation of reusable software systems is the end result of SDLC.

STLC(Software testing life cycle):

• STLC is mainly related to software testing.

• It focuses only on testing the software.

• STLC involves only five phases or steps.

• In STLC, less number of members (testers) are needed.

• In STLC, testing team(Test Lead or Test Architect) makes the plans and designs.

• Goal of STLC is to complete successful testing of software

• STLC phases are performed after SDLC phases.

• Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.

• A tested software system is the end result of STLC.

**Q.32 What is the difference between test scenarios, test cases , and test script ?**

**Answer :-**  **Test scenarios :**

* Definition > A test scenario outline what functionality needs to be tested at a high level
* A test scenario is a high level description of situation or functionality that needs to be tested. it’s a broad overview of what will be tested , without going into specific steps.
* Purpose > it guides the testing process by identifying the areas to be covered. Its like a checklist of functionality or situations to be examined.
* Level of detail > High-level , focusing on “ what “ to test .
* Example > “ test use login functionality “. Or “verify order placement process”

**Test Cases :-**

Definition > A test cases is a detailed document that outlines the steps to execute a specific test ,including the inputs , expected results , and the actual results.

* It provides a specific plan for validating a particular aspect of the application.
* Purpose > It ensures that individual functionality are validating by providing instructions for testing .
* Level of Detail > Detailed , focusing on “what” and “ how to test “
* Example >

**Test Scenario :-**

Test use login functionality

Test Case :-

* Test id – TC-login-001
* Description :- Verify successfully login with valid credentials
* Precondition :- User account exists.
* Steps :-

= Open the application’s login page.

= Enter username “test user”

= Enter password “password”

= Click the “login button “

* Expected Result :- User is redirected to the application’s homepage.
* Actual Result :- ( to be fitted after execution)
* Status :- ( to be filled after execution )

**Test Script :**

* Definition > A test script is a short program written in a programming language that automatic the execution of test cases. It’s a sequence of instructions that a computer can execute to test a part of the software.
* Purpose > It automatic the testing process making it faster and more efficient. It ensure that all steps are executed correctly and consistently.
* Level of Detail > Details the specific actions to be performed by the computer , often including input values and expected results.
* Example > A script written in python or java that automates the steps of a test case , like logging in or placing an order.

**Q.33 Explain What test plan is? What is the information that should be covered?**

**Answ**er **:-**  Test plan is a comprehensive document that outlines the approach, scope, resources, schedule, and activities required for testing a software application or system.

Key information covered in a test plan includes:

**Test Objectives:**

Clearly defined goals for testing, such as identifying bugs, ensuring functionality, or meeting performance requirements.

**Scope:**

Defines what aspects of the software or system will be tested, and what is excluded.

**Testing Approach:**

Specifies the types of testing to be performed (e.g., functional, performance, security) and the methods to be used.

**Test Environment:**

Details the hardware, software, and network configurations needed for testing.

**Test Data:**

Describes the data required for testing, including its creation and management.

**Schedule and Timeline:**

Outlines the start and end dates for testing activities, as well as key milestones.

**Roles and Responsibilities:**

Assigns specific tasks and responsibilities to team members.

**Risk Management:**

Us alI identifies potential risks and outlines mitigation strategies.

**Entry and Exit Criteria:**

Defines the conditions that must be met before testing can begin and when it can be considered complete.

**Test Deliverables:**

Specifies the documents, reports, and other artifacts that will be produced during testing.

**Tools and Technologies:**

Lists the tools used for test management, automation, and other testing activities.

**Q.34 what is priority?**

**Answer :-** In software testing, priority refers to the order in which a defect or issue should be addressed and resolved.

It indicates how quickly a bug needs to be fixed, primarily based on its impact on business objectives and user experience, rather than solely on its technical severity.

**Levels:**

Priority is commonly categorized into levels such as:

**Urgent/Critical:** Requires immediate resolution as it severely impacts the application or prevents its use.

**High**: Needs to be resolved quickly as it significantly affects functionality or user experience.

**Medium:** Should be addressed in upcoming builds but does not require immediate stoppage of work.

**Low:** Can be fixed after higher-priority issues are resolved.

**Q.35 What is Severity?**

**Answer:-**Severity is defined as the extent to which a particular defect can create an impact on the software. Severity is a parameter to denote the implication and the impact of the defect on the functionality of the software.

A higher effect of the bug on system functionality will lead to a higher severity level.

QA engineer determines the severity level of a bug.

**Types of Severity:**

Severity in software testing can be classified into 4 categories:

**Critical:** - This severity level implies that the process has been completely shut off and no further action can be taken.

**Major: -** This is a significant flaw that causes the system to fail. However, certain parts of the system remain functional.

**Medium:-** This flaw results in unfavorable behavior but the system remains functioning.

**Low:-** This type of flaw won't cause any major breakdown in the system.

**Q.36 Difference between severity and priority ?**

**Answer:- Severity (Technical Impact)**

**Definition:** How serious the defect is from the system’s functionality point of view.

**Focus:** Measures the impact of the bug on the application.

**Set by:** Tester (since testers analyze functionality).

**Examples:**

**High Severity:** Login button not working → users can’t log in.

**Low Severity:** A small spelling mistake in the "Help" section.

**Priority (Business Impact)**

**Definition:** How urgently the defect needs to be fixed.

**Focus:** Measures the order in which the bug should be fixed according to business needs.

**Set by:** Product Manager / Business / Client (sometimes Dev lead).

**Examples:**

**High Priority:** Wrong price shown on a product page during a sale.

**Low Priority:** Logo colour slightly different from design.

**Q.37 Explain the difference between Authorization and Authentication in web testing, what are the common problem faced In web testing?**

**Answer :-** Authentication and Authorization are distinct security process often confused

Authentication verify a users identify

**Ex :-** Login with a username and password

Authorization determination what resources that authenticated user is allowed to access

**EX :-** Granting permission to view a file

Essentially authentication proves who you are and authorization defines what you can do .

**Q.38 What are the difference Methodologies in agile development model?**

**Answer:-**

**Agile Methodology:**

**1. Scrum master:**

**2. extreme programing:**

**3. Kanban:**

**1) Scrum master:**

This framework uses short, iterative cycles called sprints, typically lasting 2-4 weeks, to deliver working software in increments.

Scrum emphasizes teamwork, roles like Scrum Master and Product Owner, and regular feedback through sprint reviews and retrospectives.

**2) extreme programing:**

XP emphasizes technical excellence and rapid delivery, with practices like pair programming, test-driven development, and frequent releases.

It prioritizes close collaboration and feedback from customers.

**3) Kanban:**

Kanban focuses on visualizing workflow and limiting work in progress (WIP) to optimize throughput and identify bottlenecks.

It's a more flexible approach than Scrum, suitable for continuous delivery and ongoing development.

**Q.39 Create HLR and test cases of WhatsApp web ?**

**Answer:-**

**HLR of WhatsApp web**

**1. Login & Authentication**

The user must be able to log in by scanning a QR code from their mobile WhatsApp app.

Session should sync with the mobile device.

Auto-login should work until user logs out manually.

**2. Messaging**

Users should be able to send and receive text messages in real time.

Users should be able to send/receive images, videos, documents, voice notes.

Emoji, stickers, and GIFs must be supported.

**3. Contacts & Chats**

User should see all existing chats from mobile.

User should be able to search contacts and chats.

User should be able to create new groups and view group chats.

**4. Notifications**

Desktop/browser notifications should appear for new messages.

Unread message counts should be displayed.

**5. Multimedia Handling**

Users should be able to download and view shared media.

Users should be able to play voice notes and videos.

**6. Security**

End-to-end encryption must be maintained.

Auto log-out on device disconnection or inactivity.

**7. Settings & Profile**

User should be able to change profile picture, status, and about.

Notifications, themes (dark/light mode) should be available.

**8. Performance & Compatibility**

WhatsApp Web must work smoothly across supported browsers (Chrome, Edge, Firefox, Safari).

Must support multiple platforms (Windows, Mac, Linux).

**👉 Sample Test Cases for WhatsApp Web**

**1. Login & Authentication**

**TC\_01:** Verify user can log in by scanning the QR code.

**TC\_02:** Verify login fails if QR code is expired.

**TC\_03:** Verify session remains active after refreshing the browser.

**TC\_04:** Verify user is logged out when mobile internet is disconnected.

**2. Messaging**

**TC\_05:** Verify sending a text message to a contact.

**TC\_06:** Verify receiving a text message from a contact.

**TC\_07:** Verify sending an image, video, document, and voice note.

**TC\_08:** Verify sending emojis, stickers, and GIFs.

**TC\_09:** Verify the message delivery ticks (✓ sent, ✓✓ delivered, ✓✓ blue for read).

**3. Contacts & Chats**

**TC\_10:** Verify user can search a contact by name.

**TC\_11:** Verify user can open an existing chat.

**TC\_12:** Verify user can create a group.

**TC\_13:** Verify user can see group participants and group info.

**4. Notifications**

**TC\_14:** Verify desktop notifications appear for new messages.

**TC\_15:** Verify unread message count appears correctly on chats.

**TC\_16:** Verify mute notification option works for a chat/group.

**5. Multimedia Handling**

**TC\_17:** Verify user can download received images/videos.

**TC\_18:** Verify user can play received voice notes.

**TC\_19:** Verify user can play received videos.

**6. Security**

**TC\_20**: Verify end-to-end encryption message appears in chat window.

**TC\_21**: Verify user is auto-logged out after inactivity (if session expired).

**TC\_22**: Verify logging out from phone logs out from WhatsApp Web.

**7. Settings & Profile**

**TC\_23:** Verify user can change profile picture.

**TC\_24:** Verify user can change About/Status.

**TC\_25:** Verify dark/light mode works correctly.

**8. Performance & Compatibility**

**TC\_26:** Verify WhatsApp Web works on Chrome browser.

**TC\_27:** Verify WhatsApp Web works on Firefox browser.

**TC\_28:** Verify WhatsApp Web works smoothly with multiple open chats.

**Q.40 write test cases on WhatsApp chat message ?**

**Answer :-**  User need to create a Whatsapp Ac

user can select a another user ac in the contact list to chat with.

verify that user can send a text message to another user.

verify that user can receive a text message to another user.

verify that user can send media file to another user .

verify that user can receive media file to another user .

user can send images , video , audio and documents.

user can receive the images , video , audio and documents.

verify that user send live location and current location to another user .

verify that user can receive the location to send another user .

verify that user create a pool and send to the another user.

verify that user receive a pool for given to the opinion.

user can create an AI images and send to the another user.

verify that user can receive the AI images to send by another user.

user can send event schedule to another user.

user can receive the event schedule by sanded to another user.

verify that user can send emojis , stickers , Avtar and gifs send to the another user.

verify that user receive the emojis , stickers , Avtar and gifs by sanded to the another user .

verify that user can send voice notes can be recorded , play and send .

verify that user can receiver voice notes and can be play

verify that type indicator { "typing..." } when show to other user is typing.

verify that read receipts ( double blue tick ) are displayed correctly.

verify that messages can be delete for me & delete for everyone show in display.

verify that long messages are wrapped or scrollable.

verify that chat time stamps are displayed correctly.

verify that search functionality inside chats.

verify that pinned, chat copy , info, star , Forward and replay function are show in display.

verify that view contact , new group , media , link , docs, mute notification , disappearing messages ,and chat theme functionality shows in display.

verify that more options shows Report , block , clear chat , export chat , add shortcut , add to list functionality available .

verify that camera and payment option shows in a chat functionality .

verify that user can do the video and audio call to the another user.

verify that user can receive the video and audio call by the another user.

**NEGATIVE SCENARIOS**

verify sending message without an internet connection ( message should show pending status )

verify that large file sizes ( beyond WhatsApp limit ) can not be sent

verify sending unsupported file formats.

verify behavior when the recipients has blocked the sender .

verify behavior when storage is full on the device.

**Q.41 Write down test scenario of pen .**

**Answer:- Test scenario of pen**

**1. Functional Scenarios**

Verify that the pen writes smoothly on paper.

Verify that ink flows consistently without gaps.

Verify that the pen starts writing immediately without shaking or scribbling first.

Verify that the pen writes in the intended ink color (blue, black, red, etc.).

Verify that the pen can write on different paper types (rough, smooth, glossy).

Verify the click or cap mechanism works as expected.

**2. Usability Scenarios**

Verify that the grip is comfortable for different hand sizes.

Verify that the weight of the pen is comfortable for long writing sessions.

Verify the pen can be used easily by both left-handed and right-handed users.

**3. Durability Scenarios**

Verify the pen continues to work after being dropped from table height.

Verify the clip (if present) remains intact after repeated use.

Verify the pen works after being kept unused for a long time.

Verify the pen cover is hard material when dropped from table height.

**4. Ink & Refill Scenarios**

Verify the pen writes for the expected length before running

Out of ink.

Verify ink does not leak under normal storage conditions.

Verify that the refill can be replaced (for refillable pens)

**5. Environmental Scenarios**

Verify the pen writes in varying temperature conditions (hot, cold).

**Negative scenario**

👉 If the pen writes on wet paper

Verify the function of a pen when a user tries to write on unsupported surfaces like glass , plastic , wood , etc.

If the pen length is too long its difficult to balance for writing

When the point is broken ink should start from likage

If the pen refill spring Is broken pen can not click

When the plastic thread is worn out pen is not close properly

**Q.42** **write down test scenario of pen stand .**

**Answer :- Test scenario of pen stand : -**

**1 . Functional Scenarios**

Verify that the pen stand can hold the expected number of pens/pencils.

Verify that the pens do not fall out easily when placed in the stand.

Verify that different sizes of pens/pencils fit properly.

Verify that other small stationery items (markers, scissors, rulers) can fit if applicable.

Verify stability — the pen stand does not tip over easily when loaded.

**2. Usability Scenarios**

Verify that the opening of the pen stand is wide enough for easy access.

Verify that pens can be removed and placed back without obstruction.

Verify that the height of the pen stand allows users to see and grab items easily.

**3. Durability Scenarios**

Verify that the pen stand can withstand a fall from table height without damage.

Verify that the surface does not crack or chip under normal usage.

Verify that the pen stand maintains its shape after prolonged use.

1. **Aesthetic & Design Scenarios**

Verify that the color, finish, and design match product specifications.

Verify that there are no sharp edges or rough surfaces that could harm the user.

Verify that branding or printed labels are clear and not peeling.

**5. Environmental Scenario**

Verify that the pen stand does not get damaged in varying temperature conditions.

Verify that it is resistant to minor water spills (if applicable).

**Q.43** **Write down test scenario of chair ?**

**Answer:- Test Scenarios for a Chair**

1. **Functional Scenarios**

Verify that the chair can support the specified maximum weight capacity.

Verify that all legs of the chair are stable and level on the floor.

Verify that the chair height matches product specifications.

Verify that the chair’s backrest is properly aligned and secure.

Verify that armrests (if present) are firmly

attached and comfortable.

**2. Usability Scenarios**

Verify that the chair is comfortable to sit on for extended periods.

Verify that the seat height and depth are suitable for different user heights.

Verify that the chair is easy to move around (lightweight or with wheels if applicable).

Verify that the chair design allows for good posture.

**3. Durability Scenarios**

Verify that the chair can withstand continuous

sitting for long durations without damage.

Verify that screws, joints, or welds do not loosen over time.

Verify that the chair remains stable after repeated use.

Verify that the seat cushion (if present) retains shape after prolonged use.

**4. Safety Scenario.**

Verify that the chair has no sharp edges or protruding parts.

Verify that all parts are securely fixed to avoid accidents.

Verify that the chair does not tip over easily when leaning forward or backward.

**5. Aesthetic & Design Scenarios**

Verify that the chairs color, material, and finish match the design specifications.

Verify that stitching or upholstery (if present) is neat and intact.

Verify that the design is consistent across multiple units in bulk production.

**Q.44 Write a Test scenario of ATM.**

**Answer:-**

1.Verify that the ATM accepts valid ATM/debit cards.

2. Verify that the ATM rejects invalid/blocked/expired cards.

3. Verify that the ATM asks for a PIN after card insertion.

4. Verify that a user can check account balance.

5. Verify that a user can withdraw cash within available balance.

6. Verify that the ATM dispenses correct denomination and amount.

7. Verify that a user can print a mini-statement/receipt.

8. Verify that a user can transfer funds (if supported).

9. Verify that a user can change PIN successfully.

10. Verify that the ATM returns the card after transaction.

11. Verify that the ATM logs out automatically after inactivity.

13. Verify that PIN is masked (hidden) when entered.

14. Verify that ATM supports multiple bank cards (inter-bank transactions).

15. Verify that surveillance camera is active during transactions.

16. Verify ATM works during power failure (backup generator/UPS).

\*\*NEGATIVE SCENARIOS \*\*

17. Verify entering an incorrect PIN (should deny after 3 attempts & block card).

18. Verify behavior when requested amount > account balance.

19. Verify behavior when requested amount > ATM cash availability.

20. Verify behavior when requested amount exceeds daily withdrawal limit.

21. Verify behavior when card is inserted upside down/damaged.

22. Verify cash not dispensed but account debited (should raise complaint).

23. Verify behavior when receipt printer runs out of paper.

24. Verify behavior when cash tray is jammed.

25. Verify behavior when ATM is out of service.

26. Verify timeout scenario when user leaves session idle.

**Q.45 When to use usability testing?**

**Answer:-** User testing should be conducted early and often throughout the design and development process.

It's beneficial at various stages, from initial concept testing and prototyping to pre-launch and even post launch for ongoing improvements.

**Key benefits of testing throughout the process:- Reduced development costs:**

Identifying and fixing issues early is much cheaper than making changes later.

**Improved user experience:**

User feedback directly informs design decisions, leading to a more intuitive and user-friendly product.

**Increased user satisfaction:**

By addressing user needs and pain points, user testing can lead to higher levels of user satisfaction.

**Reduced risk of failure:** Testing helps ensure that the product meets user needs and expectations before launch, reducing the risk of a failed product launch.

**Q.46 What is the procedure for GUI Testing?**

**Answer:-**

The procedure for Graphical User Interface (GUI) testing typically involves a structured approach to ensure the visual elements and functionalities of an application's interface meet specified requirements and provide a positive user experience.

**Test Planning and Design:**

Define the scope of GUI testing, including the specific areas and elements of the UI to be tested.

Identify target environments (e.g., different operating systems, browsers, screen resolutions) for compatibility testing.

Determine testing objectives and success criteria.

Select appropriate testing techniques (manual, automated, or a combination).

**Test Case Development:**

Create detailed test cases for each GUI element and interaction, covering both positive and negative scenarios.

Include steps to verify layout, design, functionality, usability, and responsiveness.

Ensure test cases address aspects like element size, position, alignment, font readability, color schemes, image clarity, error message display, and navigation.

Separate test data from test cases for reusability and maintainability.

**Test Execution:**

Execute the developed test cases manually or using automated testing tools.

Manually inspect visual elements and interactions for accuracy and adherence to design specifications.

Utilize automation tools to simulate user interactions and perform repetitive checks efficiently.

Record test results, including any deviations or defects observed during execution.

**Defect Reporting and Tracking:**

Document any identified bugs or issues with clear descriptions, steps to reproduce, and expected versus actual results.

Prioritize defects based on severity and impact.

Track the status of reported defects until they are resolved and re-tested.

**Reporting and Analysis:**

Generate reports summarizing the GUI testing results, including test coverage, defect trends, and overall quality of the interface.

Analyze the findings to identify areas for improvement in the GUI design and development process.

**Q.47 Teat Scenarios of microwave oven**

**Answer :-**

1. Verify that the microwave turns on when plugged in and powered.

2. Verify that the door opens and closes properly.

3. Verify that the microwave does not start if the door is open.

4. Verify that the microwave starts heating when Start button is pressed.

5. Verify that the microwave stops heating when Stop/Cancel button is pressed.

6. Verify that the microwave runs for the set cooking time.

7. Verify that the timer countdown works correctly.

8. Verify that the microwave light turns on while heating.

9. Verify that the turntable rotates while heating.

10. Verify that the microwave beeps when cooking is completed.

11. Verify that preset modes (e.g., Defrost, Popcorn, Reheat) work properly.

12. Verify that the power level setting works (Low, Medium, High).

13. Verify that the clock and timer can be set.

14. Verify that the microwave retains settings after power cut (if supported).

15. Verify that child lock function works (microwave should not start).

16. Verify that buttons/controls are responsive.

17. Verify that the display screen shows correct cooking time and settings.

18. Verify that the labels on buttons are clear and readable.

19. Verify that the door handle is easy to grip and open/close.

20. Verify that interior is illuminated during cooking for visibility.

21. Verify that the microwave heats food evenly.

22. Verify that microwave maintains temperature accuracy for different power levels.

23. Verify that the outer body does not overheat during long usage.

24. Verify that the microwave stops if overheating is detected.

25. Verify that the microwave complies with radiation leakage stander

**NEGATIVE SCENARIOS**

26. Verify that the microwave does not start when no food item is inside (some models block this).

27. Verify behavior when a metal utensil is placed inside (should warn/error).

28. Verify behavior when a plastic container not microwave-safe is used.

29. Verify behavior when cooking time is set to 0 seconds (should not start).

30. Verify setting an invalid time input (e.g., 99:99).

31. Verify microwave does not exceed maximum timer limit.

32. Verify what happens if the door is opened while running (should stop heating immediately).

33. Verify continuous running for long time (overheating protection).

34. Verify that the microwave does not start if Start button is pressed repeatedly too fast.

35. Verify that the microwave does not restart automatically after power outage (safety check).

**Q.48 write Scenario of Coffee vending machine?**

**Answer :-**  • Verify that the machine powers on and initializes correctly.

• Verify that the user can select a type of beverage (e.g., Coffee, Tea, Espresso).

• Verify that the user can choose beverage customization options (e.g., sugar level, milk, strength).

• Verify that the machine displays the correct price for each selected beverage.

• Verify that the machine accepts all supported payment methods (e.g., coins, notes, card, UPI)

• Verify the machine dispenses the correct beverage after successful payment

• Verify that the machine returns the correct balance if extra payment is inserted

• Verify that the machine does not start the dispensing process without payment

• Verify the machine handles out-of-stock items gracefully (e.g., displays “Out of Coffee”)

• Verify that the machine stops dispensing once the desired quantity is dispensed.

• Verify that the machine heats the beverage to the correct temperature.

• Verify that the beverage is dispensed into the correct container (e.g., cup holder)

• Verify that the machine handles power failure during operation without safety issues.

• Verify the refill process for coffee beans, milk, sugar, and water by operator

• Verify machine shows an error or warning if internal components malfunction (e.g., no water supply, jammed dispenser)

• Verify that multiple users can use the machine one after another without conflict.

• Verify that the machine has a cleaning cycle or notifies when cleaning is needed.

• Verify that the screen (if touch-enabled) responds accurately to user input

• Verify that invalid or foreign currency is rejected properly.

• Verify that the machine logs transactions for audit (if applicable).

**Q.49 Create Taste cases on Composes mail functionality**

**Answer:-**

**Positive Test Cases**

**1. TC01 - Open Compose Window**

Action: Click on Compose button.

Expected: New compose mail window opens.

**2. TC02 - Enter Recipient (To)**

Action: Enter a valid email address in the "To" field.

Expected: Address is accepted without error.

**3. TC03 - Enter Subject**

Action: Enter text in the Subject field.

Expected: Subject is displayed correctly.

**4. TC04 - Enter Email Body**

Action: Type message in the body.

Expected: Text is displayed properly in the body.

**5. TC05 - Add C**C

Action: Add a valid email in the CC field.

Expected: Address is accepted.

**6. TC06 - Add BCC**

Action: Add a valid email in the BCC field.

Expected: Address is accepted.

**7. TC07 - Send Mail**

Action: Fill all fields and click Send.

Expected: Mail is sent successfully, confirmation shown.

**8. TC08 - Attach File**

Action: Attach a file (e.g., PDF, DOCX, image).

Expected: File uploads successfully and is visible in mail draft.

**9. TC09 - Draft Save Automatically**

Action: Start writing mail and close without sending.

Expected: Mail is auto-saved in Drafts.

**10. TC10 - Format Text**

Action: Apply formatting (bold, italic, underline).

Expected: Formatting is applied correctly in the body.

**Negative Test Cases**

**11. TC11 - Empty Recipient**

Action: Leave To field empty and click Send.

Expected: Error message "Recipient required" is shown.

**12. TC12 - Invalid Email Format**

Action: Enter an invalid email (e.g., abc@.com) in To field.

Expected: System rejects with an error.

**13. TC13 - Oversized Attachment**

Action: Try attaching a file larger than allowed (e.g., 30MB in Gmail).

Expected: System displays error “File size too large.”

**14. TC14 - Special Characters in Subject**

Action: Enter special characters (<>#$%^&\*) in Subject.

Expected: System handles without crashing or rejects invalid characters.

**15. TC15 - Exceed Maximum Recipients**

Action: Add recipients beyond allowed limit (e.g., 500 in Gmail).

Expected: System blocks or throws warning.

**16. TC16 - Expired Session**

Action: Compose mail when session expires and click Send.

Expected: User redirected to login page.

**17. TC17 - Network Failure During Send**

Action: Try sending mail without internet connection.

Expected: Mail not sent, error displayed.

**18. TC18 - Empty Subject**

Action: Leave subject empty and click Send.

Expected: Prompt “Do you want to send without subject?”

**19. TC19 - Empty Body**

Action: Leave body empty and click Send.

Expected: Mail is still sent (if system allows).

**20. TC20 - Unsupported File Type**

Action: Attach blocked file type (e.g., .exe).

Expected: System rejects with warning.

**Q.50 To create HLR and Test cases on online shopping product ( flipkart)**

**Answer:-**

**HLR for Online Shopping (Buy Product)**

1. User Account & Login

System shall allow users to create an account and log in securely.

2. Product Browsing & Search

System shall allow users to browse categories and search products by name, brand, or filter.

3. Product Details Page

System shall display product name, price, description, reviews, and stock availability.

4. Add to Cart

System shall allow users to add products to the shopping cart.

5. View & Modify Cart

System shall allow users to update quantity, remove items, and view total price.

6. Checkout Process

System shall allow users to proceed to checkout with selected items.

7. Delivery & Address Management

System shall allow users to enter/select delivery address.

8. Payment Options

System shall support multiple payment methods (Credit/Debit Card, UPI, Wallet, COD)

9. Order Confirmation

System shall generate an order summary and confirmation message.

10. Order Tracking

System shall allow users to track order status (processing, shipped, delivered).

**Test Cases for Online Shopping (Buy Product)**

**Positive Test Cases**

1. TC01 - Login Successfully

Steps: Enter valid username & password → Click Login

Expected: User is logged in successfully.

2. TC02 - Search Product by Name

Steps: Enter product name in search bar → Search

Expected: Relevant products are displayed.

3. TC03 - View Product Details

Steps: Click on a product → Open details page

Expected: Product info (price, stock, description) shown.

4. TC04 - Add Product to Cart

Steps: Select product → Click “Add to Cart”

Expected: Product is added to cart.

5. TC05 - Update Product Quantity in Cart

Steps: Increase quantity in cart → Update

Expected: Cart updates with new quantity and total price.

6. TC06 - Checkout Successfully

Steps: Proceed to checkout → Select address → Payment → Confirm

Expected: Order placed successfully, confirmation shown.

7. TC07 - Pay with Credit Card

Steps: Choose “Credit Card” → Enter valid details → Pay

Expected: Payment successful, order confirmed.

8. TC08 - Cash on Delivery (COD)

Steps: Select COD → Confirm order

Expected: Order confirmed with COD option.

9. TC09 - Order Confirmation Email

Steps: Place order successfully

Expected: Confirmation email/SMS is sent.

10. TC10 - Track Order Status

Steps: Go to “My Orders” → Select order

Expected: Status shown (e.g., “Shipped”).

***Negative Test Cases***

11. TC11 - Login with Wrong Password

Expected: Error message “Invalid username or password.”

12. TC12 - Search Invalid Product

Expected: “No results found” message displayed.

13. TC13 - Add Out-of-Stock Product

Expected: System prevents adding and shows “Out of Stock.”

14. TC14 - Invalid Quantity in Cart

Steps: Enter quantity = 0 or negative

Expected: System rejects with an error.

15. TC15 - Checkout without Address

Expected: Prompt “Please enter delivery address.”

16. TC16 - Payment Failed (Invalid Card)

Expected: Error “Payment failed, try again.”

17. TC17 - Exceed Payment Timeout

Expected: Order not placed, session timeout error shown.

18. TC18 - Network Disconnect During Checkout

Expected: System shows error and does not confirm order.

19. TC19 - Multiple Tabs Checkout Conflict

Expected: Only one valid order placed, duplicate prevented.

20. TC20 - Cancel Order

Steps: Try cancelling after order shipped

Expected: System rejects cancellation.

**Q.51 Write a Scenarios of wrist watch ?**

**Answer :-**

1. Verify that the watch shows the current time correctly.
2. Verify that the watch can display hours, minutes, and seconds.
3. Verify that the watch supports 12-hour and 24-hour format (if digital).
4. Verify that the date and day are displayed correctly.
5. Verify that the watch can be manually adjusted for time/date.
6. Verify that the watch supports alarm setting and rings at the correct time.
7. Verify that the stopwatch/timer works correctly (start, stop, reset).
8. Verify that the watch resets properly after battery replacement.
9. Verify behavior when time is set to 11:59 → 12:00 (AM/PM switch).
10. Verify behavior when time is set to 23:59 → 00:00 (24-hour switch).
11. Verify that invalid dates (e.g., Feb 30, Apr 31) are not accepted.
12. Verify the watch’s accuracy when time is running for long hours (drift check).
13. Verify that the alarm does not ring when disabled.

**🔹 Usability Scenarios**

Verify that the watch display is readable in normal light.

1. Verify that the backlight/illumination works in the dark (digital).
2. Verify that the watch hands (analog) are visible and properly aligned.
3. Verify that the watch is comfortable to wear with different strap sizes.
4. Verify that the buttons/knobs are easy to operate.
5. Verify that the watch is water resistant up to its rated depth.
6. Verify that the glass is scratch resistant.
7. Verify the watch’s performance under temperature variations (hot/cold).
8. Verify the watch’s battery life as per specifications.
9. Verify that the strap and buckle are strong and durable.

**Q.52 Write test scenario of lift elevator?**

**Answer:-**

1. Verify that the lift moves up when a higher floor is selected.
2. Verify that the lift moves down when a lower floor is selected.
3. Verify that the lift stops at the correct floor
4. Verify that the lift doors open automatically when it reaches a selected floor.
5. Verify that the lift doors close after a few seconds if no one enters/exits.
6. Verify that the open door button works while the lift is at a floor.
7. Verify that the close door button works while the lift is at a floor.
8. Verify that the lift does not move when doors are open.
9. Verify that the lift moves only when a valid floor number is pressed.
10. Verify that the lift ignores duplicate floor requests (if already pressed).
11. Verify that the lift does not move beyond top floor or below ground floor.
12. Verify behavior when the lift is overloaded (alarm/door does not close).
13. Verify behavior when a power failure occurs (emergency light & stop).
14. Verify that the emergency stop button halts the lift immediately.
15. Verify that pressing a non-existent filth or number (e.g., 13 if skipped) is ignored.
16. Verify that the sensor prevents door from closing workingis in between.
17. Verify that the lift has an emergency alarm button workinthe g properly.
18. Vthe erify that floor indicators show the correct current floor.
19. Verify that the direction indicator (up/down arrow) is correct.
20. Verify that the voice/announcement system announces the correct floor.
21. Verify time taken for the lift to reach from ground to top floor.
22. Verify door open/close timing is within expected range.
23. Verify lift can handlthe e multiple requests from different floors.
24. Verify that waiting time for the lift is acceptable under loproperly

**Q.54 write the test scenario of Whatsapp group chat ?**

**A.** Verify that a user can create a group chat with multiple participants .

⦁ verify that user can send a text messages in a group chat.

⦁ verify that user receive a text messages in a group chat.

⦁ verify that all group members can receive the message.

⦁ verify that all group members can send the message.

⦁ verify that the group name and group icon can be set or changed.

⦁ verify that group description can be set / edited.

⦁ verify that the members can see typing indicator { "typing..." } of other members.

⦁ verify that read receipts ( double blue ticks ) work per member in group.

⦁ verify that messages deleted for everyone disappears for all group members.

⦁ verify that admin can add new participants.

⦁ verify that admin can remove participants.

⦁ verify that group cam have multiple admins ( make / remove admin )

⦁ verify that all group member can send images , documents , AI images , video , and voice notes in a group.

⦁ verify that all group member can receive a messages images , documents , AI images , video and voice notes in a group.

⦁ verify that all group members send location, Event schedule , contact , payment and pool.

⦁ verify that all group members can received location , Event schedule , contact , payment and pool .

⦁ verify that pools and reactions work properly in group chats.

⦁ verify that all group members can audio and video calling together .

⦁ verify that in group chat setting only admin have access to send the text messages and media files. another members have no access.

⦁ verify end - to - end encryption works in group chats .

⦁ verify that group info , group media, search , mute notifications , disappearing messages , chat theme functionality are shown in a display .

⦁ verify that more options report ,exit group , clear chat , export chat , add to shortcut , add list are work properly.

**NEGATIVE SCENARIO**

⦁ Verify when a non - admin members tries to add and remove to members ( should not be allowed )

⦁ verify sending a message in a group without internet ( messages should show pending status )

⦁ verify when a user is removed from the group they should not send / receive new messages.

⦁ verify when a group reaches maximum participants limit ( cannot add more )

⦁ verify when blocked contact is in the same group ( you should not receive private messages , but group messages should still be visible)

⦁ verify behavior when a user leaves the group ( should display " user left " notification ).

⦁ verify that muted group do not send notifications.

**Q.55 Whatsapp payment scenario?**

**A.**

1. Verify that a user can set up WhatsApp Payments by linking a bank account.

2. Verify that the UPI ID is generated/linked correctly.

3. Verify that a user can send money to a contact using WhatsApp Payments.

4. Verify that a user can request money from another contact.

5. Verify that money sent is debited from sender’s account and credited to recipient’s account.

6. Verify that transaction status (Success, Pending, Failed) is shown properly.

7. Verify that transaction history is available in the Payments section.

8. Verify that bank balance check option works correctly.

9. Verify that QR code payments can be scanned and processed.

10. Verify that payment notifications are sent to both sender and receiver.

11. Verify that the Payments option appears in WhatsApp settings.

12. Verify that the linked bank name, last 4 digits, and UPI ID are displayed correctly.

13. Verify that the payment confirmation screen shows correct amount, recipient, and account.

14. Verify that the UPI PIN input is masked (hidden for security).

15. Verify that the transaction receipt contains transaction ID, bank reference ID, date, and time.

16. Verify that end-to-end encryption applies to payment transactions.

17. Verify that timeout handling works if the transaction takes too long.

18. Verify that WhatsApp does not store UPI PIN or sensitive bank details.

19. Verify that fraudulent/unauthorized transactions are blocked.

20. Verify that payments work correctly under low network conditions

**\*\*NEGATIVES SCENARIOS\* \***

21. Verify that a payment fails if the wrong UPI PIN is entered.

22. Verify that a payment fails if the bank server is down.

23. Verify that a payment fails if the user has insufficient balance.

24. Verify that a user cannot make a payment without linking a bank account.

25. Verify behavior when sending to a non-WhatsApp Payment user (should prompt alternate options).

26. Verify behavior when the internet is disconnected during a transaction.

27. Verify that a payment fails if the transaction amount exceeds UPI limit.

28 . Verify that duplicate transactions are not processed if a user clicks Pay multiple times.

29. Verify that a blocked contact cannot send/request payment.

20. Verify what happens if the recipient deletes WhatsApp account before payment is processed.