**Software Testing**

**(Module – 2) Fundamental**

**Assignment – 2**

**NAME: - Dhruvin Sathwara**

* **What is Exploratory Testing?**

Exploratory Testing is the simultaneous learning, Test design, and Test Execution process

In this Test planning, analysis, design, and execution are all done instantly together

It is the type of software testing in which the tester is free to select any possible methodology to test the software. It is an unscripted approach to software testing where tester is not aware with system.

In exploratory testing the system is randomly given to the tester and Tester will do all tasks at a time when system given.The focus of exploratory testing is more on testing as an “Exploring” activity.

* **What is Traceability Matrix?**

**Requirement Traceability matrix** is a document that links system requirements with the test cases.

The main purpose of the requirement traceability matrix is to verify that the all requirements of clients are covered in the test cases designed by the testers. Traceability Matrix will be in Table form

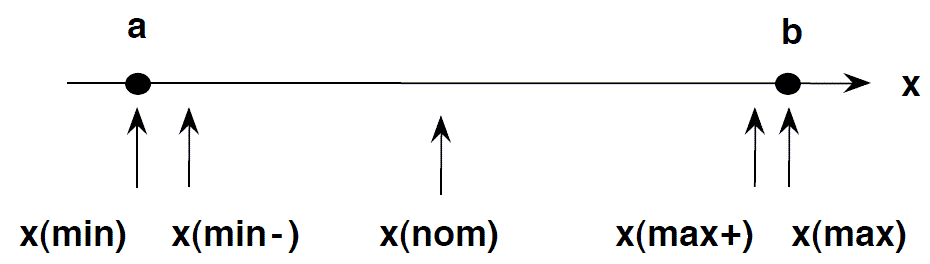
In this software requirements provided by the client have to be further split into different scenarios and further to test cases. Each of this test-case has to be executed individually. The tested test cases for relevant requirement are marked with “Right” in table which gives you the proper tracing.

* **What is Boundary Value Testing?**

[Boundary Value Analysis](https://www.geeksforgeeks.org/boundary-value-analysis-triangle-problem/) is based on testing the boundary values of valid and invalid partitions. The behaviour at the edge of the equivalence partition is more likely to be incorrect than the behaviour within the partition.

It checks for the input values near the boundary that have a higher chance of error.

Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition.



Inputs will be selected as the Minimum value, just above the minimum value, The Nominal value, Maximum Value, Just Below the maximum.

* **What is Equivalence Partitioning Testing**

It is the Black Box testing Technique.

In Equivalence Partitioning the input data of software unit is divided into the partitions of equivalent data from which test cases can be derived

The test cases are designed to cover each partition at least one. It reduces time required for testing because of small number of test cases.

If a condition of one partition is true, then the condition of another equal partition must also be true, and if a condition of one partition is false, then the condition of another equal partition must also be false. It is applied when there is a range of input values

Ex: - Valid Range and Invalid Range of inputs

The six digit OTP number. The less than 6 digit and more than 6 digit OTP will be Invalid range and 6 digit OTP will be Valid input range.

* **What is Integration Testing?**

It is the level of software testing process where individual units or components are combined and tested as a group. Testing is performed to expose the defects in the interfaces and in the interactions between integrated units or components.

The purpose is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in integration testing.

It tests interaction between components, interactions to different part of the system such as operating system, file system and hardware.

There are two types of method of integration testing. 1) Big bang integration Testing

2) Incremental Integration Testing

- Top Down Approach

- Bottom Up Approach

* **What is Alpha Testing?**

Alpha testing is the type of User acceptance testing. Alpha testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public.

It is always performed by the developer at the software development site. Some time it is also performed by the individual Testing Team.

It comes under the category both White box testing and Black box testing. It is always performed in virtual environment within the organization.

We can say that the software is not open to the market and public while it is in Alpha Testing phase.

* **What is Beta Testing?**

Beta testing is the process of testing a software product or system in a real-world environment before its official release. It helps to identify bugs and errors that may have been missed during the development process.

Beta testing is performed by real users of the software application in real environment. It is one of the types of User Acceptance Testing.

It is always performed outside of the organization and always open to the market and public. It is only kind of black box testing.

* **What is Component Testing?**

Component is the smallest testable part of the software. Component testing is the level of software testing process where individual components of the software are tested. The purpose is to validate each component of software performs as designed.

The frameworks, drivers, stubs and fake objects are used to assist in component testing.

It is typically written and run by the software developers to ensure that code meets its design and behave as intended. It is performed with white box testing method.

The goal of component testing is to isolate each part of program and show that individual parts are correct.

* **What is functional system testing?**

Functional system testing means a requirement that specifies a function that a system or system component must perform.

Each functionality of the software application is tested by providing appropriate test input, expecting the output, and comparing the actual output with the expected output.

There is two types of Test Approach

1. Requirement Based Functional Testing:-

Testing involves validating a software system based on specified requirements and specifications.

1. Process Based Testing:-

Testing should reflect the business environment and processes in which the system will operate.

Therefore, test cases should be based on real business processes.

* **What is non-functional testing?**

Non-functional testing is a type of software testing that is performed to verify the non-functional requirements of the application. It is the testing of “how” the system works. Non-functional testing may be performed at all test levels.

It tests all the aspects that are not tested in functional testing. It checks the non-functional attributes of the system.

In this testing the performance, usability, compatibility, security of the system are tested.

* **What is GUI testing?**

It is the testing of systems Graphical User Interface under test.

GUI testing is the process for ensuring proper functionality of the graphical user interface (GUI) for a specific application. GUI testing generally evaluates a design of elements such as layout, colours and also fonts, font sizes, labels, text boxes, text formatting, buttons, lists, icons, links etc.

It allows you to test the functionality from a user’s perspective. Sometimes the internal functions of the system work correctly but the user interface doesn’t then GUI testing is good.

* **What is Adhoc Testing?**

Adhoc testing is also called the Error Testing. Main aim of this testing is to find defects by random checking.

It is the informal testing type with an aim to break the system. It is done by the tester whose testing knowledge is high. It does not follow any testing techniques to create test cases.

Adhoc Testing does not follow any structured way of testing and it is randomly done on any part of application.

Types of Adhoc testing:

1. Buddy testing: - In this two buddy works mutually together on same module to identify defects. In this one will be Developer and other will be Tester. This testing usually happens after unit testing completion.
2. Pair Testing: - In this testing two Testers are assigned on same module. The one will execute test cases and another will note down it.
3. Monkey Testing: - It is the Testing technique of randomly test the product or application without test cases with goal to break the system

* **What is load testing?**

It is the performance testing to check system behaviour under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.

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

Example: - An Airline website was not able to handle 10000+ users during a festival offer.

Load Testing helps identify the bottlenecks in the system under heavy user load. Load testing will determine whether system needs to be fine-tuned or modification of hardware and software is required to improve performance.

* **What is Stress Testing?**

Stress testing is to determine the limit, at which the system or software or hardware breaks.

The goal is to see the robustness and ability of an application to not crash and recover successfully after heavy load. The goal of stress testing is to analyse the behaviour of the system after failure.

With Stress testing, we aim to find a breaking point of the application with testing by over data. It should able to display appropriate error message when the system is under stress.

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

It is also called the Structure based Technique or Glass box Testing.

White box Testing is based on an analysis of the internal structure of the component or system. Here the testers require knowledge of how the software is implemented and how it works.

The white box testing is done by the developers who have knowledge of Coding. White box testing is the detailed investigation of internal logic and structure of the code.

To perform white box testing on an application, the tester needs knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit of the code is behaving inappropriately.

There are testing techniques as below: -

1. Statement Coverage
2. Branch Coverage.
3. Condition Coverage.
4. Multiple Condition Coverage.
5. Basis Path Testing.
6. Loop Testing.

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

The black box testing is the testing of Functional or non-functional without knowing internal structure of the system. It is also called input/output driven testing.

The testers have no knowledge of how the system or component is structured inside the box. They view the software as a black box with inputs & outputs.

It is what a system does, rather than HOW it does. The tester is not aware to the system architecture and they do not have the access to the source code.

In black box testing, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked.

There are different testing techniques.

1. Equivalence partitioning.
2. Boundary value analysis.
3. Decision tables.
4. State transition testing.

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

The defect is the occurrence of variance between the Expected result and the Actual result.

Defect is some kind of error or mistake which prevents the system to work smooth.

There is different type of categories.

1. Database defect: - It deals with improper handling of data in the database. Ex. Values not inserted or deleted in database properly.
2. Critical Functionality defect: - The occurrence of this bug hampers the crucial functionality of the application. Ex. Main functionality, Exception.
3. Functionality defect: - This defect affects the functionality of the application. Ex. Button like save, delete not performing their intended function. Missing Functionality.
4. Security defect: - Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix. Ex. Authentication: - accepting an invalid username and password. Authorization: - Accessibility to pages through permission not given.
5. User Interface defect: - As the name suggests, the bugs deal with problems related to UI are usually considered less severe. Ex. Spelling mistake, Alignment problem, improper error/message.

* **Mention what Bigbang testing is?**

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

Developer develops the individual modules one by one and then all modules are gathered at a time then tester tests it as a whole system. Big Bang testing has the advantage that everything is finished before integration testing starts.

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

Exit criteria are the defined requirements in software testing that must be met in order to determine that testing has been completed.

Exit criteria is used to determine whether a given test activity has been completed or NOT.

Software testing teams will use exit criteria to determine if a test plan or project can exit to the next stage. Exit criteria can be defined for all of the test activities right from planning, specification and execution.

* **When should "Regression Testing" be performed?**

Regression testing should be done to ensure that existing functionality works as expected after making changes to code, such as adding new features, fixing bugs of previously tested program, or improving performance.

It should be done to ensure that, after changes made in module it does not affecting the other unchanged areas. It helps to avoid introducing new defects while modifying the system.

* **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

1. Testing shows presence of Defects: -

It can say that defects are present in software, but cannot say that there is no defects are present. Software testing reduces the presence of defects. The goal of software testing is to make the software fail. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects.

2. Exhaustive Testing is Impossible: -

Testing everything including all combinations of inputs and preconditions is not possible. It is not possible to test each and every field of software. We have to test only accordingly to the risk based data or randomly. It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case.

3. Early Testing: -

It should be started very early from the beginning as soon as requirements are gathered. The defect detected in the early phases of SDLC will be very less expensive. For better performance of software, software testing will start at the requirement analysis phase of software development life cycle.

4. Defect Clustering: -

It means the group of defects. In a project, a small number of modules can contain most of the defects. It means we do not have to test the whole system, but the only defected part and nearest area of defected part.

5. Pesticide Paradox: -

If the same tests are repeated over and over again, at some point the test-case will be no longer able to find new bugs. It means that we have to make changes in test-cases with different inputs to find new bugs. Testers cannot simply depend on existing test techniques. He must look out continually to improve the existing methods to make testing more effective.

6. Testing is context dependent: -

The testing approach depends on the context of the software developed. Different context types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the Android application.

7. Absence of Errors Fallacy: -

It shows that no software testing will be 100% error free. If the system built is unusable and does not fulfil the user’s needs and expectations then finding and fixing defects does not help.

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

|  |  |  |
| --- | --- | --- |
| **Quality Assurance** | **Quality Control** | **Testing** |
| Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. | Activities which ensure the verification of developed software with respect to documented 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. |
| 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. |
| It is responsible for the entire software development life cycle. | It is responsible for the software testing life cycle. | It is responsible for the software testing life cycle. |
| The aim of quality assurance is to prevent defects. | The aim of quality control is to identify and improve the defects. | The aim of testing is to finding defects from system |
| It works with the development process | It’s done after the development process | This usually happens after the software has been created |

* **Difference between Smoke and Sanity?**

|  |  |
| --- | --- |
| **Smoke** | **Sanity** |
| [Smoke testing](https://www.geeksforgeeks.org/smoke-testing-software-testing/) is done to assure that the critical or main functionality of program is working fine. | [Sanity testing](https://www.geeksforgeeks.org/sanity-testing-software-testing/) is done to check the bugs have been fixed after the build and no further issues are introduced due to these changes. |
| Smoke testing is performed by either developers or testers. | Sanity testing is normally performed by testers. |
| Smoke testing is documented. | Sanity testing isn’t documented. |
| Smoke testing is also called subset of acceptance testing. | Sanity testing is also called subset of regression testing. |
| Smoke testing is done to measure the stability of the system | Sanity testing is done to measure the rationality of the system |
| It includes all the system’s essential basic functionality. | It includes only those modules where change in code is made. |
| Smoke testing is performed when new product is built. | Sanity testing is conducted after the completion of regression testing. |

* **Difference between Verification and Validation.**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| Focuses on whether the software meets its specified requirements | Focuses on whether the software meets the needs of end-users |
| Helps to ensure that the software is built correctly | Helps to ensure that the correct software is built |
| It is conducted during the development process | It is conducted after the development process |
| It includes checking documents, design, codes and programs. | It includes testing and validating the actual product. |
| Verification is the static testing. | Validation is the dynamic testing. |
| Verification is done by the QA Team. | Validation is done by the Tester Team. |

* **Explain types of Performance testing.**

1. Load Testing: - It is the performance testing to check system behaviour 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.

2. Stress Testing: - Stress testing is to determine the limit, at which the system or software or hardware breaks.

The goal is to see the robustness and ability of an application to not crash and recover successfully after heavy load.

3. Endurance testing: - Endurance testing is also known as ‘Soak Testing’. It is done to determine if the system can sustain the continuous expected load for a long duration.

It helps identify any issues that may occur after prolonged usage of the system.

4. Spike testing: - In spike testing, we analyse the behaviour of the system on suddenly increasing the number of users. It also involves checking if the application is able to recover after the sudden burst of users.

5. Volume testing: - The volume testing is performed by feeding the application with a high volume of data. The application can be tested with a large amount of data inserted in the database processing. Using volume testing, we can identify the bottleneck in the application with a high volume of data.

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

Error: - An error in software development is a mistake that a software developer makes when writing code. When the developer fails to understand a requirement definition and hence that misunderstanding gets translated into buggy code then error happens.

Defect: - A defect refers to a situation when the application is not working as per the requirement and the actual and expected result of the software is not matching. It is the error which found by the tester then it is called the Defect.

Bug: - A bug refers to defect which means that the software product is not working as per the requirements set. When we have any type of logical error, it causes our code to break, which results in a bug. A defect accepted by the developer then it becomes the Bug.

Failure: - Events when software doesn’t perform as expected during execution called the Failure. It indicates the presence of defects. Failure is detected by end-users once they face a particular issue in the software.

* **What is Bug Life Cycle?**

Mainly bug life cycle refers to its entire state or time span starting from a new defect detected to the closing off of that defect by the tester.

The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various stages and make the defect fixing process systematic. Bug Status in defect life cycle is the present state from which the defect or a bug is currently undergoing.

Defect status is current state or progress of a defect or bug to track and understand the actual progress of the defect life cycle.

* **Explain the difference between Functional testing and Non-Functional testing**

|  |  |
| --- | --- |
| **Functional Testing** | **Non-Functional Testing** |
| Functional testing is performed using functional specification provided by the client and verifies the system against the functional requirements. | Non-functional testing is performed for the non-functional requirements which should be present in the system like performance, reliability, stability etc. |
| Functional testing is executed first. | Non-functional testing should be performed after functional 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 system does. | Non-functional testing describes how good the system works. |
| To find and fix defects in the system’s functionality. | To find and fix bottlenecks and performance issues in the system. |
| Types: Unit testing, Integration testing, system testing, and acceptance testing. | Types: Performance testing, scalability testing, usability testing, security testing. |

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

|  |  |
| --- | --- |
| **SDLC** | **STLC** |
| SDLC stands for the Software Development Life Cycle | STLC stands for the Software Testing Life Cycle |
| It focuses on entire process of software development. | It focuses on entire process of software testing. |
| It consists phases like Requirement gathering, planning, design, coding, testing, deployment. | It consists phases like Requirement analysis, Test planning, Create Test case, Execute test case, test closure. |
| Goal of SDLC is to ensure that the software is developed in a way that meets the requirements. | Goal of the STLC is to identify and document any defects or issues in the software application and resolve them. |
| The end result of SDLC is a good quality product gets delivered to the customer. | The outcome of STLC is ideally to deliver defect free software. |
| SDLC phases are completed before the STLC phases. | STLC phases are performed after SDLC phases. |

* **To create HLR & Test case of 1) (Instagram , Face book) only first page. 2) Facebook login page:**
* HLR of Instagram:-

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/HLR%20of%20Instagram%20Login%20Page.xlsx>

* Testcase of Instagram:-

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/TEST%20CASE%20-%20INSTAGRAM%20LoginPage.xlsx>

* HLR of Facebook Login:-

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/HLR%20of%20Facebook%20Login%20Page.xlsx>

* Testcase of Facebook Login:-

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/TEST%20CASE%20-%20FACEBOOK%20Login%20Page.xlsx>

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

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Test Case** | **Test script** |
| Test scenario is the functionality that can be tested | Test case is the set of actions to be executed to verify particular functionality | It is the set of instructions to test an app automatically using programming |
| It is derived from the Business requirements and software requirements | It is mostly derived from the Test Scenarios | It is mostly derived from the test cases |
| Helps to test to end to end functionality in agile way | Helps in exhaustive testing of a system | Helps to test specific thing repeatedly |
| It is mostly focused on what to test | It is focused on what to test and how to test | It is focused on the expected result |
| It includes an end to end functionality to be tested | It includes a test steps, data, expected result for testing etc. | It includes different commands to develop a test script |
| Allows the quickly assessing the testing scope | Allows the detecting the errors and defects | Allows carrying out an automatic execution of test cases |
| The main task is to check the full functionality of a software application | The main task is to verify compliance with the standards, guidelines, and customer requirements. | The main task is to verify that nothing is skipped and results are true as the desired testing plan. |

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

A test plan is a document that outlines the objectives, scope, approach, resources, and schedule for a testing project. It defines the strategy for how testing will be conducted, including what will be tested, who will perform the tests, the test environment, and the criteria for success. The purpose of a test plan is to ensure that testing is systematic, thorough, and aligned with the project's goals and requirements.

A comprehensive test plan typically includes the following information:

**-Introduction:** Provides an overview of the document and its purpose.

**-Objectives:** Clearly states the goals and objectives of the testing effort.

**-Scope:** Defines the boundaries of the testing, including what will and will not be tested.

**-Approach:** Describes the overall strategy for testing, including methodologies, techniques, and tools to be used.

**-Test Items:** Lists the specific components or features to be tested.

**-Features to be tested:** Describes the features or functionalities to be tested, often with reference to requirements or specifications.

**-Test Environment:** Specifies the hardware, software, and other resources needed for testing.

**-Test Deliverables:** Lists the documents, reports, and other deliverables expected from the testing process.

**-Test Schedule:** Outlines the timeline for testing activities, including milestones and deadlines.

**-Test Team:** Identifies the roles and responsibilities of individuals involved in testing.

**-Risks and Assumptions:** Identifies potential risks to the testing process and any assumptions made during planning.

* **What are the different Methodologies in Agile Development Model?**
* Individuals and interactions, Over processes and tools:

Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in Agile, it is preferable to interact with client, manager or team regarding issue and make sure that the issue gets resolved.

* Working software, Over comprehensive documentation:

Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. For example, you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because, in the end, the client needs a document.

* Customer collaboration, Over contract negotiation:

Contract negotiation is important as they make the budget of software, but customer collaboration is more important than over contract negotiation. For example, if you stuck with the requirements or process, then do not go for a contract which we have negotiated. You need to interact with the customer, gather their requirements.

* Responding to change, over following a plan:

In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software.

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

**Authentication:** Authentication is the process of verifying the identity of a user or entity attempting to access a system or resource.

In web testing, this typically involves validating that a user is who they claim to be by verifying their credentials, such as username and password. Common authentication methods include username/password, biometric authentication, single sign-on (SSO), and multi-factor authentication (MFA).

**Authorization:** Authorization, on the other hand, determines what actions or resources a user is allowed to access after they have been authenticated. It is the process of granting or denying access rights and privileges to authenticated users based on their identity and the permissions associated with their account or role.

In web testing, authorization testing involves verifying that users can only access the resources and perform the actions they are authorized to, based on their role or privileges.

Common problems faced in web testing include:

**Security vulnerabilities:** Web applications are often targets for malicious attacks, so security testing is crucial to identify and address vulnerabilities.

**Compatibility issues:** Web applications need to work across various browsers, devices, and operating systems. Compatibility testing ensures that the application functions correctly and displays properly across different environments.

**Performance issues:** Poor performance, slow loading times, and high server response times can negatively impact user experience and lead to decreased engagement and customer dissatisfaction. Performance testing helps identify and address bottlenecks, scalability issues, and other performance-related problems.

**Usability concerns:** Usability testing evaluates the user interface, navigation, and overall user experience of the web application to ensure it is intuitive, user-friendly, and meets the needs of its target audience.

* **To create HLR & Test Case of Web Based (WhatsApp web , Instagram web)**
* HLR of whatsapp web

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/HLR%20of%20WHATSAPP%20WEB.xlsx>

* Testcase of Whatsapp web

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/TEST%20CASE%20-%20WHATSAPP%20WEB.xlsx>

* HLR of Instagram Web login page

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/HLR%20of%20INSTAGRAM%20WEB%20Login%20Page.xlsx>

* Testcase of Instagram web login page

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/TEST%20CASE%20-%20INSTAGRAM%20WEB%20LoginPage.xlsx>

* **To create HLR and TestCase on this Link. https://artoftesting.com/**
* HLR of ART OF TESTING site

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/HLR%20of%20ART%20OF%20TESTING.xlsx>

* Testcase of ART OF TESTING site

<https://github.com/DS1022/Assignment-2_Practicals/blob/main/TEST%20CASE%20of%20ART%20OF%20TESTING.xlsx>

* **Write a scenario of only Whatsapp chat messages**
* Verify that the Chat window contains the entire chat list.
* Verify that the Chat window displays the contact numbers whose numbers not saved in mobile.
* Verify that Chat window displayed with all contacts with DP or without DP.
* Verify that the Chat window is displayed on the group chat list.
* Verify that the Chat window displays the last updated chatting time.
* Verify that the Chat window displays the name of all contacts on the chat window.
* Verify that clicking on one Chat contact then a new window should open with history.
* Verify that the user can see all delivered and received messages.
* Verify that the user can see the read or send time of messages.
* Verify that the user can send and receive text messages in the individual chat box.
* Verify that user can send and receive documents in the individual chat box.
* Verify that user can send and receive photos in individual chat box.
* Verify that user can send and receive videos in individual chat box.
* Verify that user can send and receive Contacts in individual chat box.
* Verify that user can send and receive Location in individual chat box.
* Verify that user can send and receive GIF in individual chat box.
* Verify that user can send and receive Stickers in individual chat box.
* Verify that user can delete text, video, audio, locations, documents in individual chat box.
* Verify that user can send recorded voice mail in individual chat box.
* Verify that user can delete the entire chat history in individual chat box.
* Verify that user is able to see contact details in individual chat box.
* Verify that user is able to share images, links, and documents from media in individual chat box.
* Verify that user is able to search specific chat history using search option in individual chat box.
* Verify that user is able to video call in individual chat box.
* Verify that user is able to voice call in individual chat box.
* Verify that user is able to mute the individuals in individual chat box.
* **Write a scenario of pen.**
* Verify that user is able to hold pen comfortably.
* Verify that the pen is writing smoothly or not.
* Verify that pen is not making any type of sound while writing.
* Verify that the ink flow of the pen is smooth or not. There should not be any break in ink flow.
* Verify that the quality of the material of pen is good or bad.
* Verify that the company name of pen is visible clearly or not.
* Verify that the ink colour of the pen should be consistent from start to end.
* Verify that the pen is able to write on different types of paper. (Smooth, Glossy, rough, thick, thin.)
* Verify that the ink of pen is waterproof or not.
* Verify that the ink of pen is dried quickly on paper or not.
* Verify that other refills fit the pen or not.
* Verify that the pen shouldn't have any sharp edges and corners.
* Verify that the body or pen and the ink are made up of nontoxic material or not.
* Verify that pen is working normally or not after putting the pen in water for some time.
* Verify that pen is working normally or not after dropping the pen from some height.
* Verify that ink spelled or not after dropping the pen from some height.
* **Write a scenario of pen stand.**
* Verify that the pen stand is stable.
* Verify any visible defects or damages in the material.
* Verify that maximum number of pens or other stationery items that can be accommodated in the pen stand.
* Verify if the material is resistant to scratches, stains, or other forms of damage that may occur during regular use
* Verify Evaluate the weight distribution of the pen stand by placing pens of varying weights and sizes in different compartments.
* Verify Ensure that the pen stand remains stable and balanced, even when loaded with items unevenly.
* Verify the compatibility of the pen stand with different desk surfaces, including wood, glass, or plastic.
* Verify Ensure that the pen stand does not scratch or damage the desk surface during normal use.
* Verify Assess the ease of access to pens or other items stored in the pen stand.
* Determine how easy it is to clean the pen stand, considering factors such as material type and design intricacies.
* Verify the pen stand is resistant to water or if it requires special cleaning instructions to maintain its appearance.
* Verify the pen stand arrives without damages or defects that may have occurred during transit.
* Consider user reviews and ratings to understand overall satisfaction with the pen stand.
* **Write a scenario of door.**
* Verify the door is single door or bi-folded door.
* Verify the door opens inwards or outwards.
* Verify that the dimension of the doors is as per the specifications.
* Verify that the material used in the door body and its parts is as per the specifications.
* Verify that colour of the door is as specified.
* Verify if the door is sliding door or rotating door.
* Verify the position, quality and strength of hinges.
* Verify the type of locks in the door.
* Verify the number of locks in the door interior side or exterior side.
* Verify if the door is having peek-hole or not.
* Verify if the door is having stopper or not.
* Verify if the door closes automatically or not – spring mechanism.
* Verify if the door makes noise when opened or closed.
* Verify the door condition when used extensively with water.
* Verify the door condition in different climatic conditions- temperature, humidity etc.
* Verify the amount of force- pull or push required to open or close the door.
* Verify the handle to open/close the door, can be used by various size of human hands
* Verify that we are able to lock and unlock the door
* Verify if someone knocking the door we are able to listen or not
* Verify that someone people knocking the door we are able to see
* Verify the door is water proof or not.
* **Write a scenario of ATM.**
* Verify that power backup should be present at ATM.
* Verify that card reader should be present.
* Verify that receipt printer should be present and working.
* Verify that cash dispenser is working as expected.
* Verify that the key pad should be working and covered.
* Verify that buttons are displayed on screen of ATM machine.
* Verify the font of text on the screen, it should be clearly visible.
* Verify that when card in inserted in ATM, pin should be asked from user.
* Verify that when user enters incorrect pin for a particular number of times, the card is blocked.
* Verify that when user enters correct pin, the user details should be displayed on ATM screen.
* Verify that ATM machine asks to user for the amount to be withdrawn.
* Verify that if use enters amount greater than daily withdraw limit, error message is displayed.
* Verify that if doesn't enter amount in round off digits, error message is displayed.
* Verify that if user enters valid amount, the exact amount of cash should be dispensed from ATM machine.
* Verify that how much time is taken in a transaction.
* Verify how much time is taken by system to logout user.
* Verify that user is able to use card of other bank on the ATM.
* Verify that message is displayed when the cash in ATM is finished.
* Verify that correct message is displayed after the transaction.
* Verify that user is presented with an option to select language of operation.
* Verify that pin is displayed in masked format.
* Verify that error message is displayed when entered amount is greater than account balance.
* Verify that session is timeout is no activity is performed for a particular time.
* Verify that the user is not allowed to exceed one transaction limit amount.
* Verify that the user is not allowed to exceed the one-day transaction limit amount.
* Verify the functionality by entering a wrong pin number for a particular number of times.
* Verify the card ATM machine functionality by inserting an expired card.
* **When to used Usability Testing?**

All fields on a page (For Example, text box, radio options, drop-down lists) should be aligned properly.

The user should not be able to type in drop-down select lists.

Tab and Shift +Tab order should work properly.

All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard.

All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard,

All pages should have a title.

Confirmation messages should be displayed before performing any update or delete operation.

Hourglass should be displayed when the application is busy

Page text should be left-justified.

The user should be able to select only one radio option and any combination for checkboxes.

Goal of Usability Testing are Effectiveness of the system, Efficiency, Accuracy, User Friendliness.

* **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 barstool bar, menu bar, dialog boxes and windows etc.

WHAT DO YOU CHECK IN GUI TESTING?

Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.

Check you can execute the intended functionality of the application using the GUI.

Check Error Messages are displayed correctly.

Check for Clear demarcation of different sections on screen.

Check Font used in application is readable.

Check the alignment of the text is proper.

Check the Colour of the font and warning messages is aesthetically pleasing.

Check that the images have good clarity.

Check that the images are properly aligned.

Check the positioning of GUI elements for different screen resolution.

* **Write a scenario of microwave Owen.**
* Verify that type of microwave oven like solo, grill or convection.
* Verify that the company name is properly displayed or not.
* Verify that the Brand logo is properly displayed on the. microwave oven or not.
* Verify that size of the microwave oven.
* Verify that colour of the microwave oven.
* Verify that material of the microwave oven.
* Verify that capacity of the microwave oven.
* Verify that the compact design of the microwave oven.
* Verify that glass is turn able or not.
* Verify that weight of the microwave oven.
* Verify that dimensions of the microwave oven.
* Verify that voltage of the microwave oven.
* Verify that batteries are required or not.
* Verify that all buttons are properly worked or not.
* Verify that food is properly reheating or not.
* Verify that food is grilled properly or not.
* Verify that the digital displayed screen should be properly visible to users.
* Verify that oven's door is properly opened and get closed.
* Verify that different kind of food at different temperature.
* Verify that different kind of containers.
* Verify that temperature functionality is properly working or not.
* Verify that the alarm sound system is properly working or not.
* Verify that glass rotation speed is as expected.
* Verify oven's condition when it runs for specific hours.
* Verify that disconnecting power while cooking is in progress.
* **Write a scenario of coffee vending machine.**
  + Verify that the dimension of the coffee machine is as per the specification.
  + Verify that outer body, as well as inner part’s material, is as per the specification.
  + Verify that the machine’s body colour as well brand is correctly visible and as per specification.
  + Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder.
  + Verify that the quantity of hot water, milk, coffee powder per serving is correct.
  + Verify the power/voltage requirements of the machine.
  + 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.
  + Verify that coffee should not leak when not in operation.
  + Verify the amount of coffee served in single-serving is as per specification.
  + Verify that the digital display displays correct information.
  + Verify the machine can be switched on and off using the power buttons.
  + Verify for the indicator lights when the machine is switched on-off.
  + Verify that the functioning of all the buttons work properly when pressed.
  + Verify that each button has an image/text with it, indicating the task it performs.
  + Verify that the coffee served has the same and correct temperature each time it is served by the machine.
  + Verify that system should display an error when it runs out of ingredients.
  + Verify that pressing the coffee button multiple times leads to multiple serving of coffee.
  + Verify that there is the passage for residual/extra coffee in the machine.
  + Verify that machine should work correctly in different climatic, moistures and temperature conditions.
  + Verify that machine should not make too much sound when in operation.
  + Verify the amount of time the machine takes to serve a single serving of coffee.
  + Verify the performance of the machine when used continuously until the ingredients run out of the requirements.
  + Verify the functioning of the coffee machine when two/multiple buttons are pressed simultaneously.
  + Verify the functioning of coffee machine with a lesser or higher voltage than required.
  + Verify the functioning of the coffee machine if the ingredient container’s capacity is exceeded.
  + Verify for the cup holder dimension as per specification/ or market standard
  + Verify for guarantee & warranty of the machine, in case provided.
* **Write a scenario of chair.**
* Verify that the chair is stable enough to take an average human load.
* Verify that material used in making the chair-wood, plastic etc.
* Verify that chair’s leg are level to the floor.
* Verify that usability of the chair as an office chair, normal household chair.
* Verify there is back support in the chair.
* Verify there is support for hands in the chair.
* Verify the paint’s type and color.
* Verify if the chair’s material is brittle or not.
* Verify cushion is provided with chair or not.
* Verify condition when washed with water or effect of water on chair.
* Verify that the dimension of chair is as per the specifications.
* Verify that the weight of the chair is as per the specifications.
* Verify the height of the chair’s seat from floor.
* Verify whether distance between all legs it should be same.
* Verify by design because there is a chairs with 3 legs or modern design when two and two are different distance.
* Verify that wheels are needed specified in chair.
* Verify the wheels are turning 360 degrees or not.
* Verify the wheels are oiled well or not.
* Verify that chair can be upgraded (different back) or add the wheels.
* Verify chair can be stored with other chairs on top/bottom.
* Verify that strength and material of the chair.
* Verify that the Main Function check stability.
* Verify edged of chair is needed to be in round shape or sharp.
* **To create scenario (positive & negative) 1. Gmail (Receiving mail) 2. Online shopping to buy product (flipkart)**

**1. Gmail (receiving mail):**

* + Verify that a newly received email is displayed as highlighted in the Inbox section.
  + Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line).
  + Verify that the user receives the email in their inbox.
  + Verify that on clicking the newly received email, the user is navigated to email content.
  + Verify that the email contents are correctly displayed with the desired source formatting.
  + Verify that any attachments are attached to the email and are downloadable.
  + Verify that the attachments are scanned for viruses before download.
  + Verify that all the emails marked as read are not highlighted.
  + Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.
  + Verify that count of unread emails is displayed alongside ‘Inbox’ text in the left sidebar of Gmail.
  + Verify that unread email count increases by one on receiving a new email.
  + Verify that unread email count decreases by one on reading an email (marking an email as read).
  + Verify that email recipients in cc are visible to all users.
  + Verify that email recipients in bcc are not visible to the user.
  + Verify that all received emails get piled up in the ‘Inbox’ section and get deleted in cyclic fashion based on the size availability.
  + Verify that email can be received from non-Gmail email Ids like yahoo, Hotmail etc.

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

* + Verify the initiation of the buy flow.
  + Verify the accuracy of product details.
  + Verify responsiveness to changes in quantity.
  + Verify the behaviour with out-of-stock items.
  + Verify the adding products from different categories.
  + Verify the accuracy and function of applied discounts.
  + Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.
  + Verify that users can add products to the wishlist.
  + Verify that the user can see the previously added products on the cart page, after signing in to the application.
  + Verify that the user can successfully buy more than one products that were added to his/her cart.
  + Verify that the user cannot add more than the available inventory of the product.
  + Verify that the limit to the number of products a user can buy is working correctly. Also, an error message gets displayed, preventing the user from buying more than the limit.
  + Verify that the delivery can be declined during checkout for the places where shipping is not available.
  + Verify that the Cash on Delivery option of payment is working fine.
  + Verify that the different prepaid methods of payments are working fine.
  + Verify that product return functionality works correctly
  + Verify the visibility and accuracy of delivery options.
  + Verify responsiveness of the payment selection step.
  + Verify the accuracy of order summary details.
  + Verify responsiveness to changes in user location.
  + Verify the redirection to the Order Confirmation page.
  + Verify the cancelling the payment process midway.
* **Write a scenario of wrist watch.**
* Verify the type of watch – analog or digital.
* Verify the correctness time displayed by the second, minute, and hour hand of the watch.
* Verify the digital display for hours, minutes, and seconds is correctly displayed.
* Verify the material of the watch and its strap.
* Verify the shape of the dial is as per specification.
* Verify the dimension of the watch is as per the specification.
* Verify the weight of the watch.
* Verify the watch is waterproof or not.
* Verify that the numbers in the dial are clearly visible or not.
* Verify the watch is having a date and day display or not.
* Verify the colour of the text displayed in the watch – time, day, date, and other information.
* Verify that clock’s time can be corrected using the key in case of an analog clock and buttons in case of a digital clock.
* Verify the second hand of the watch makes ticking sound or not.
* Verify the brand of the watch and check if its visible in the dial.
* Verify the clock is having stopwatch, timers, and alarm functionality or not.
* Verify the format of the watch 12 hours or 24 hours.
* Verify the watch comes with any guarantee or warranty.
* Verify the dial has glass covering or plastic, check if the material is breakable or not.
* Verify the dial’s glass/plastic is resistant to minor scratches or not.
* Verify the battery requirement of the watch.
* Verify belt or chain used is comfortable or not and it’s length
* Verify chain material and belt for damage.
* **Write a scenario of lift(Elevator)**
* Verify the maximum passenger capacity of lift.
* Verify the maximum weight capacity of lift.
* Verify the Height, Width and length of lift.
* Verify the type of material used in lift interior and exterior.
* Verify the type of door of lift.
* Verify that buttons are present according to the number of floors in building.
* Verify that buttons to open and close lift door should be present.
* Verify that the buttons should be self-explanatory.
* Verify that blind person should be able to use the buttons.
* Verify that the controls are guided by a voice message.
* Verify that emergency button and contact details should be present in lift.
* Verify the performance of lift, the time taken to go a particular floor.
* Verify that fan is present in lift or not.
* Verify that light is present in lift or not.
* Verify that lift moves to particular floor when the floor button is pressed.
* Verify that lift comes to floor is up/down button are pressed at a particular floor.
* Verify that the door should not be opened if open is door button is pressed in between two floors.
* Verify the behaviour of lift in case of power failure, lift should not free fall or should not stop in between two floors.
* Verify that lift door closes or not if an object is placed in between the door, it should not close.
* Verify the time for which the door remains open at floor.
* Verify that door should not open while the lift is in motion.
* Verify that backup mechanism should be present in case of power loss.
* Verify that in case multiple floor number button is clicked, lift should stop at each floor.
* Verify that in case of capacity limit is reached users are prompted with warning alert.
* Verify that inside lift user are prompted with current floor and direction information the lift is moving towards.
* Verify the behaviour of lift when the capacity of lift exceeds.
* Verify the behaviour of lift when there is smoke or fire inside the lift.
* Verify the behaviour by pressing open button while lift is moving.
* Verify the behaviour by pressing stop button before reaching the specific floor.
* **Write a scenario of whatsapp Group (generate group)**
* Verify that user is able to create a new or not.
* Verify that user is able to add multiple contacts from contact list.
* Verify that user is able to insert group name and select image for DP.
* Verify that user is able to add and remove contacts from group.
* Verify that user is able to delete a group.
* Verify that user can send and receive text messages in group.
* Verify that user can send and receive documents in group chat box.
* Verify that user can send and receive photos in group chat box.
* Verify that user can send and receive videos in in group chat box.
* Verify that user can send and receive audio in in group chat box.
* Verify that user can send and receive emotions icons in in group chat box.
* Verify that user can send and receive Contacts in group chat box.
* Verify that user can send and receive Location in group chat box.
* Verify that user can send and receive GIF in in group chat box.
* Verify that user can send and receive Stickers in group chat box.
* Verify that user can delete text, video, audio, locations, documents in group chat box.
* Verify that user can send recorded voice mail in group chat box.
* Verify that user is able make multiple video call in group chat box.
* Verify that user is able to see the group contact information from Group Info in group chat box.
* Verify that user is able to shared images, links, and documents from Group Media in group chat box.
* Verify that user is able to search specific chat history using search option in group chat box.
* Verify that user is able to mute the group in group chat box.
* **Write a scenario of whatsapp payment.**
* Verify that users can register for WhatsApp Payment.
* Verifying the user's phone number for whatsapp payment.
* Verify that users can link their bank account to WhatsApp for transactions.
* Verify that sending money from one user to another within the WhatsApp application.
* Verify that users can select a contact from their address book to send money to.
* Verify that users can enter the amount they wish to send and add a note if necessary.
* Verify that users can review the transaction details before confirming.
* Verify that receiving money from another user.
* Verify that users receive a notification when they have received money.
* Verify that the transaction details are visible within the chat interface.
* Verify that users can view their transaction history.
* Verify that transaction history includes details such as date, time, amount, sender/receiver, and transaction status (completed, pending, failed, etc.).
* Verify that security features such as PIN or biometric authentication before making a payment.
* Verify that users receive confirmation messages for each transaction, including details about the recipient and amount sent.
* Verify that adding a bank account for transactions.
* Verify that users can add multiple bank accounts if supported.
* Verify that users can edit or remove bank accounts as needed.
* Verify that there are limits on the amount of money that can be sent or received per transaction and per day
* Verify that performance of the payment feature under various network conditions (3G, 4G, 5G, Wi-Fi).
* Verify that transactions are processed promptly and reliably without delays or errors.