AUTOMATION TESTING -HC FACETS WITH UFT

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

By

Neeraj Gupta (171343)

UNDER THE GUIDANCE OF

Cognizant

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May, 2021

TABLE OF CONTENTS

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled **"Automation Testing-HC Facets with UFT"** submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of my work carried out under the supervision of Cognizant**.** I have not submitted this work elsewhere for any other degree or diploma.

Neeraj Gupta

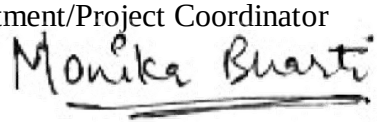
171343

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Cognizant

Date: 24/05/2021

Head of the Department/Project Coordinator



Project Report Undertaking

I **Mr.** /Ms. Neeraj Gupta, Roll No. 171343, Branch Computer Science Engineering is doing my internship with Cognizant from 27 February 2021 to 11 June 2021.

As per the procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Name: Neeraj Gupta

Roll No.: 171343

Date: 24/05/2021

ACKNOWLEDGEMENT

My success of completion of my project required guidance from many individuals and assistance from many people and I am extremely privileged to have got this all along the completion of this project.

I take this opportunity to express my gratitude to our supervisor Dr. Himanshu Jindal, for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his constant encouragement and advice throughout our project.

The in-house facilities provided by the department throughout the project are also equally acknowledgeable.

LIST OF ACRONYMS AND ABBREVIATIONS

UFT	Unified Functional Testing
QTP	Quick Test Professional
SQL	Structured Query Language
VBScript	Visual Basic Script
XML	Extensible Markup Language
JSON	Java Script Object Notation
IE	Internet Explorer
SDLC	Software Development Life cycle
QA	Quality Assurance

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Micro Focus UFT One is instrument that utilizes computerize tests to discover issues of a test application. Unified Functional Testing (UFT) is an abbreviation for "Bound together Functional Testing." It was recently known as QTP (Quick Test Professional).

Utilitarian, relapse, and administration testing are the most widely recognized utilizations for UFT. You may utilize UFT to robotize client exercises on a site or worker PC program, and afterward test and recognize surrenders for numerous clients, different informational collections, various Windows working frameworks, and additionally various gadgets utilizing similar activities. When contrasted with manual testing, computerization through UFT may save a great deal of time and cash if very much planned and done.

Today, UFT is perhaps the most generally used business mechanized testing arrangements accessible. It's notable for its effortlessness of utilization and merchant support, just as a major local area of mechanization specialists. Accordingly, qualified UFT experts have consistently been popular.

CHAPTER 1
FUNCTIONAL TESTING

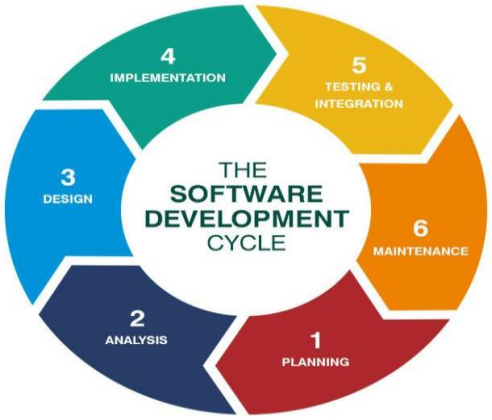
Software testing is a strategy for assessing program quality and bringing down the danger of programming mistake while being used. Most of people have experienced programming that doesn't proceed as arranged. Programming that doesn't perform as expected can cause an assortment of issues, including monetary misfortune, lost time, and harm to an organization's image. Passing or injury

Test Execution is a piece of programming testing. Programming testing is a multi-step measure that includes an assortment of assignments. One of these activities is execution. Programming testing is an innovation for finding if the program matches determined rules and affirming that it is sans blunder. It utilizes mechanization or manual testing for getting to numerous characteristics of interest by dealing with programming segments. Rather than genuine prerequisites, programming's testing will likely discover missteps, holes, and legitimate mistakes.

1.1 Software Development Life Cycle (SDLC)

The Life Cycle for Software Development (SDLC) is an interaction for programming plan, improvement and testing. The SDLC offers top notch items that fulfill customer necessities and are finished on schedule and on time. The SDLC is a strategy for a product project inside a product firm. It is a finished strategy to show how explicit programming might be assembled, overseen, refreshed and improved. The existence cycle is a method for improving the nature of programming and the improvement process.

The numerous means of a common SDLC are portrayed graphically in the outline underneath.


4
IMPLEMENTATION
5
TESTING &
INTEGRATION
3
DESIGN
THE
SOFTWARE
DEVELOPMENT
CYCLE
6
MAINTENANCE
2
ANALYSIS
1
PLANNING

Synotive

Figure1.1 Software Development Life Cycle

A typical Software Development Life Cycle consists of the following stages −

1 Planning

Examination of necessities is the most huge and principal phase of the SDLC. It is finished by the top individuals from the group with client criticism, outreach groups, market studies and experts from the business. These information are then used to characterize the center venture plan and complete item research drives in the financial, business and innovative fields.

2 Defining Requirements

The requests of the item should be fittingly determined, recorded and approved by the client or venture supervisors following the necessity examination. This is accomplished by utilizing a SRS (Software Requirement Specifications) record that incorporates all venture results to be set up and created through the entire task life cycle.

3 Designing the Product Architecture

For item designers needing to assemble the best engineering for another item, SRS is an asset. In a DDS – Specification of the plan report, various options in contrast to the idea recommended are typically given and systematized by the standards set out in the SRS.

All key partners evaluate this DDS and select an item plan procedure dependent on a few highlights, for example, hazard the executives, item steadiness, plan adaptability, spending plan and timescales.

4 Building or Developing the Product

At this period of the SDLC the real advancement of the item starts. The programming code is created as per DDS at this stage. At the point when the plan is done precisely and underlying, code age might be done quickly and without any problem.

5 Testing the Product

Since testing in present day SDLC plans is principally consolidated in all periods of SDLC, this stage is frequently an all-stage subset. This stage by and by concerns simply the testing period of the item, during which issues are accounted for, observed, fixed and checked until the item satisfies the SRS quality necessities.

6. Deployment in the Market and Maintenance

When the item is totally tried and prepared to send, it is authoritatively distributed on the applicable market. Contingent upon the plan of action of the firm, item carry out frequently happens in stages. Contingent upon input, the item would then be able to be delivered as or with recommended upgrades in the market bunch. It is kept up for the current client base after the items have been delivered to the market.

1.2 Software Testing Objectives

Figure1.2 Testing Objectives

To evaluate the work products

Before the engineer gets the work things for improvement, like the Requirements determination, Designs, and User Stories, they ought to be checked. Perceiving any vulnerability or clashing prerequisites now saves a ton of time during advancement and testing. Before the code incorporates/is prepared for testing, it goes through static examination (audits, walk-throughs, assessments, etc). Confirmation is the term for this kind of assessment. During the advancement period of any work item, it is the way toward investigating the item.

To verify requirements:

This objective exhibits that perhaps the main parts of testing is to fulfill the needs of the customer.

Analyzers look at the item and guarantee that the entirety of the specified guidelines are met. Planning each experiments, regardless of testing approach, ensures that each performed experiment is practically confirmed. A necessity detectability network (RTM) ought to likewise be made by the analyzer

to ensure that all experiments are planned to prerequisites. RTM is a useful asset for guaranteeing that experiments cover the entirety of the prerequisites.

To validate the test object:

Testing ensures the execution of necessities just as the confirmation that they work as expected by clients. Approval is the term for this sort of testing. It is the way toward assessing an item after it has been created. Approval may be done physically or consequently. It as often as possible uses an assortment of testing draws near, like Black Box, White Box, etc. Approval is frequently performed by analyzers, despite the fact that clients can likewise approve the item as a component of User adequacy testing. The buyer is viewed as the lord in any firm.

To build confidence:

Improved programming quality is quite possibly the main objectives of programming testing. There are less defects in excellent programming. To put it another way, the more compelling the testing interaction is, the less blames will be found in the last yield. Thus, the general nature of the test thing will improve. Magnificent quality prompts higher consumer loyalty and less upkeep costs.

To prevent defects:

One of the objectives of programming testing is to keep away from issues right off the bat in the advancement interaction. The cost and exertion of identifying issues early sets aside a great deal of cash and time. Imperfection anticipation involves doing a main driver of recently found issues and afterward finding a way explicit ways to stay away from comparative blunders from happening later on. Powerful testing helps in the making of a blunder free item. At the point when flaws are forestalled, the general imperfection include in the item is decreased, bringing about a greater item for the purchaser.

To find defects in product:

Another significant objective of programming testing is to discover all imperfections in an item. The objective of testing is to uncover however many imperfections as could be expected under the circumstances in a product item while likewise guaranteeing that the application addresses the client's issues. Deformities ought to be found as right on time as practical in the testing cycle.

To share information to stakeholders:

The objective of testing is to give partners the entirety of the data they need concerning specialized or different imperatives, hazard factors, confounding prerequisites, etc. It may appear as testing exercises or testing results that feature what's missing and what wasn't right.

To reduce the level of risk:

1.3 Hazard is a word which intends to the opportunity from losing. Objective of this testing is for bringing down the opportunity of a danger happening. On the off chance that we don't deal with these vulnerabilities, potential dangers will emerge during the advancement time frame, yet additionally all through the item's entire life cycle.

1.4 Benefits of Software Testing

- Cost-Effectiveness: This is the main advantages of programming testing. Testing on schedule of any task permits us to save cost later on. It is more affordable to address issues found from the get-go in the product testing measure.
- Security: Software testing's generally helpless and delicate benefit is security. Individuals are looking for things that they can trust. It helps in the early identification of perils and issues.
- Product quality: Any product item should meet this basis. Testing ensures that purchasers get a top notch item.
- Customer Satisfaction: The essential objective of each item is to give consumer loyalty. The ideal client experience is guaranteed through UI/UX testing.

1.5 Test Process in Software Testing

1) Testing is all the more a system than a solitary undertaking. Testing should be arranged, and finishing it requires discipline. The nature of the test technique utilized decides the quality and viability of programming testing. The testing interaction might be separated into the accompanying essential advances:

2) Analysis and Design

- The following are the critical errands of test examination and test plan:
- To go through the testing premise once more. The test base is alluded to as information concerning experiments like necessities, exhaustive plan, item hazard investigation, engineering and interfaces.

- Test conditions to be resolved
- Tests are made
- Planning the test climate and recognizing the innovations and devices vital.

3) Implementation and Execution

4) Evaluating Exit criteria and Reporting

Leave standards are utilized to decide when testing ought to be halted. It is dictated by code inclusion, usefulness, and hazard. All in all, it is influenced by market dangers, cost, and idealness, and differs from one undertaking to another. At the point when the accompanying conditions are met, leave measures are utilized:

- A specific level of experiments are finished with a particular pass rate.
- The bug rate falls under a specific limit
- At the point when we comply with our time constraints

5) Test Closure activities:

At the point when the product is fit to be conveyed, test conclusion tasks are finished. Testing can be ended for an assortment of reasons, including:

- the crossing out of an undertaking
- the accomplishment of a specific objective
- When there is an upkeep discharge or an update

Coming up next are the critical obligations of test conclusion exercises:

- Check that all planned expectations have been given and that every single usefulness in reports have been tended to.

- Giving over the test gear to the support group. They will help with the product.
- To survey the testing interaction and draw exercises for future deliveries and undertakings.

1.6 Levels of Testing

1.Unit testing: decides if programming segments proceed true to form.

2.Integration Testing: analyzes the progression of information starting with one module then onto the next.

3.System Testing: evaluates the two kinds that is utilitarian and non-useful testing .

4.Acceptance Testing: Verifies that a determination or agreement's models are fulfilled as per its conveyance.

Every one of these layers of testing has an unmistakable reason. The product advancement lifecycle profits by these degrees of testing.

1) Unit testing: Unit is the individual segment of a framework or program that can be assembled, stacked, and run. This kind of testing permits every module to be tried autonomously.

The objective is to test every segment of the product independently. It decides if the parts are playing out their capacities. Engineers are the ones who do this kind of testing.

2)Integration testing: The expression "mix" alludes to the way toward blending at least two , like separate programming modules are joined and after that they are tried in a gathering in this testing step to guarantee that the associated framework is prepared for testing stage. Analyzers are the ones that do this sort of testing.

3)System testing: This is done on a practical, incorporated framework. It empowers for the confirmation of the framework's consistence with the principles. It inspects the general interaction of the different segments. Burden, proficiency, soundness, and security testing are all important for the interaction. Framework testing is regularly the last advance during the time spent guaranteeing that the framework fits the necessities. It surveys both useful and non-practical testing necessities.

4)Acceptance testing: This sort of testing utilizes legitimate strides for deciding whether the standards of a determination or agreement have been fulfilled as of the date of conveyance. The client or client is

the person who does acknowledgment testing. Different investors, then again, might be dynamic all the while.

Kinds of acknowledgment testing:

Alpha testing:

This is another kind of testing that searches for programming's imperfections and blemishes. This sort of test is done towards the finish of the application advancement, before the item is dispatched or conveyed to the client, to check that the client or customer gets a mistake free programming program. Alpha testing precedes beta testing, consequently you'll have to run beta testing whenever you've finished alpha testing.

Alpha testing doesn't occur in a true setting. Maybe, these evaluations are completed in a virtual climate that intently looks like the real world.

Beta testing :

Beta testing follows alpha testing, as recently expressed. Before an item is delivered, it is exposed to beta testing. It is done in a genuine setting by few enlisted clients and clients to guarantee that the program is sans blunder and runs appropriately. A few enhancements are performed to improve the program subsequent to gathering input and productive analysis from those clients. Subsequently, while programming is in beta testing, it is alluded to as a beta form. After this round of testing is finished, the program is made accessible to the overall population.

1.7 Types of testing

1.7.1. FUNCTIONAL TESTING

It is a kind of programming testing where the product framework is approved against the useful determination or necessities. Practical tests are utilized to approve the yield of a product program by giving sufficient information and contrasting it with the useful prerequisites.

Objective of this testing

- Fundamental Usability: It involves essential convenience testing of the framework.

- Mainline Functionality: It involves testing the essential elements of an application. It assists the analyzer with finding if a client could peruse across the screens absolutely or with no issue.
- Accessibility: It affirms that the framework is effectively open to the client.
- Error Conditions: Error conditions are checked utilizing testing strategies. It inspects if the fitting blunder messages are introduced.

1.6.2 NON-FUNCTIONAL TESTING

It is a sort of programming test that explores the non-useful components of a product program (execution, ease of use, constancy, etc).

Destinations of this testing

- Nonfunctional testing should assist with improving the program convenience, effectiveness, practicality, and conveyability.
- Assists in bringing down a danger or cost of non-useful pieces of the item all through creation.
- Improve the technique wherein the item is downloaded, designed, run and kept up
- Gain a superior comprehension of how items act and what innovations are being used.

Qualities of Non-practical testing

- This testing should be quantitative, hence emotional qualities like great, better, best, etc should not be utilized.
- Exact figures are probably not going to be found at the start of the prerequisite interaction.
- Prioritizing every single need is basic.
- Ascertain that quality ascribes are precisely perceived in Software Engineering.

1.6.3 Black Box Testing

BlackBox Testing is an assessing programming strategy which incorporates testing programming applications functionalities without understanding the code structure, execution subtleties or inner pathways. Discovery Testing is a kind of programming testing which centers around programming items yields and inputs and is completely determined by programming details. Another name for it is social trying.

Figure1.3 Black Box testing

Black Box Testing Techniques

- **Proportionality Class Partition:** This method is planned to keep the measure of plausible experiments to a base while yet guaranteeing satisfactory inclusion.
- **Limit Value Analysis:** BVA likewise checks the qualities at the edges. This method surveys if the scope of qualities is satisfactory to the framework. It's an extraordinary method to eliminate the measure of experiments. It's best for frameworks with inputs that fall between explicit reaches.
- **Choice Table:** A choice table is a network that associates causes and results. Every section has an alternate blend.
-

1.6.4 White Box Testing

This is another kind of testing which is a product survey technique that contains testing the fundamental construction, engineering and code of the item to assess the progression of info and improve the plan, convenience and security. Clear box testing, open box testing, straightforward box testing, code-based testing, and glass box testing is now and again known as white box testing since the code is noticeable to analyzers.

Due to the transparent box idea, the epithet "White Box" was authored. The expression "clear box" or "White Box" alludes to the capacity to investigate the product's internal operations by means of its outside shell (or "box"). Essentially, the "black box" in "Discovery Testing" signifies the powerlessness to notice the product's internal functions, permitting just the end-client experience to be surveyed.

Confirmation of this testing

• Vulnerabilities in inside security

• Breaking or severely planned coding ways

•The stream of specific sources of info through the code

•Expected yield

•Conditional circle usefulness

• Each assertion, article, and capacity should be tried independently.

White Box Testing Techniques

Code Coverage Analysis is a well known White box testing approach. A Test Case suite's openings are filled through Code Coverage investigation. It recognizes portions of a program that aren't scrutinized in an assortment of experiments.

Code inclusion investigation might be performed utilizing robotized advances. A crate analyzer can utilize the accompanying inclusion investigation strategies:

Articulation Coverage: During the testing of code , this system commands that each possible assertion present in the code should be tried in any event once.

Branch Coverage - This technique analyzes each possible way of a product application (if-else and other restrictive circles).

Benefits of White Box Testing

•Optimizing the code by identifying covered up flaws.

•Test situations for white boxes are not difficult to robotize.

•Testing is generally more broad since each code way is covered.

•Tests may begin from the get-go in the SDLC regardless of whether a GUI isn't accessible.

Weaknesses of White Box Testing

•Testing the white box may require some serious energy and cash.

•Developers used to test situations of white box hate it. The absence of data from engineers in the testing of white cases may prompt challenges underway.

•Professional assets are needed for white box testing with a strong comprehension of programming and execution.

•Testing white box requires some investment; greater application programming needs more opportunity for complete testing.

1.6.5 Dynamic Testing

Dynamic testing is a kind of programming testing that inspects the unique conduct of programming. The fundamental objective of dynamic testing is to uncover imperfections in the program runtime climate by assessing programming conduct with dynamic factors or factors that are not consistent. To test the powerful conduct, the code should be run.

Dynamic testing Goals

The essential objective of this sort of testing is to check if the product capacities precisely both when introducing it, bringing about a dependable program liberated from extreme issues. The significant objective of the unique test is to ensure that the program is reliable. Consistency identifies with an assortment of prerequisites like execution, convenience, combativity, etc, making Dynamic Testing incredibly essential.

Dynamic Testing Advantages

•This sort of testing could unveil unseen blemishes that are excessively intense or complex to be caught by static testing.

•In this sort of testing, we run the program from starting till the end, guaranteeing code without blunder and hence improving the presentation of the product.

•This sort of testing has gotten basic for spotting security dangers.

Disadvantages of Dynamic Testing

•This sort of testing takes tremendous measure of time since it runs the application/programming or code, that burns-through a ton of resources.

•Because this sort of testing don't start right on time in the lifecycle of programming, any mistakes that are rectified later in the process may raise the expense of the venture/item.

1.6.6 Static testing

This kind of testing is a product testing procedure that searches for mistakes without running the code in the program. It is mostly utilized in getting shortcomings in the beginning of advancement measure, when they are simpler to spot and address. It likewise helps the analyzer in the discovery of deficiencies that Dynamic Testing will be unable to distinguish. Dynamic Testing, then again, analyzes an application while the code is running. These are essential methods of static testing:

•Testing program physically: Tests done physically, regularly called as surveys, incorporate manual code examination.

•Tool-helped robotized investigation: Tool-helped computerized examination is basically static investigation.

Execution of Static Testing

1.Validation of Use Cases Requirements: This guarantees that all end-client exercises, just as any information/yield, are recognized. The experiments can be more precise and complete if the utilization cases are more express and complete.

2.Validation of Functional Requirements: This guarantees that all significant perspectives are recognized in the Functional Requirements.

3.Architecture Review: This incorporates all business-level cycles like worker areas, network graphs, convention particulars, load adjusting, data set openness, test hardware, etc.

4.Validation of Prototypes/Screen Mockups: This progression involves the check of and use cases.

1.6.7 Confirmation testing

Affirmation testing is a kind of programming testing where analyzers retest a product item to guarantee that recently revealed absconds have been fixed or are not, at this point present. At the point when a test fizzles, analyzers for the most part report a deformity. Testing group will then retest to check whether the detailed issue has been tended to. Affirmation testing is the term for this. Re-testing is another term for affirmation testing. To copy the issue, analyzers should allude to the imperfection report that was arranged when the issue was first detailed. The imperfection report helps the analyzer in playing out the test by guaranteeing that a similar test measures, test information, and test climate are utilized.

Execution of Confirmation Testing

Affirmation testing is completed under the accompanying circumstances inside the product testing life cycle.

1. Whenever the advancement group delivers another form with bug fixes

2. Verify the bug fix prior to doing relapse testing.

3. When the improvement group dismisses an issue report from an analyzer. To repeat the issue, analyzers do affirmation testing.

1.6.8 Regression testing

It is a kind of programming testing utilized for guaranteeing that a program or code alteration hasn't broken current functionalities. Relapse Testing is only a total or incomplete re-execution of recently performed experiments to affirm that current usefulness is working appropriately. This testing guarantees that new code adjustments don't have potentially negative results for current usefulness. It ensures that the old code keeps on working after the latest code adjustments have been made.

Need of Regression Testing

This sort of testing arises regularly when there is a need to modify the given program and furthermore on the off chance that we need to check whether the changed program impacts various parts in the given

program. It is additionally required when an engineer adds or changes highlight of a program, just as for bug and execution issues.

Relapse Testing Tools

Relapse testing costs will rise if your product is regularly refreshed. Testing the application physically builds the hour of execution of test and costs in such conditions. In such

examples, robotizing the product utilizing this apparatuses is the most ideal choice. The measure of robotization is controlled by the quantity of experiments that might be reused in ensuing relapse cycles.

1.Selenium: It is the opensource device for web application robotization. Selenium is a program based relapse testing apparatus.

2.Quick Test Professional (QTP): QTP is a robotized programming that might be utilized to computerize practical and relapse test situations. This computerizes utilizing VBScript programming. It's a watchword driven, information driven instrument.

3.Rational Functional Tester (RFT): This is a Java device for robotizing programming application test situations. It interfaces with Rational Test Manager and is generally utilized for computerizing relapse test situations.

Difficulties in Regression Testing:

The most widely recognized relapse issues:

•Test suites fill in size as more relapse runs are performed. The entire relapse test suite can't be arrived behind schedule to time and monetary limitations.

•Reducing the test situations while keeping up most extreme measure of inclusion is as yet the issue.

•Determining the recurrence of these Tests, like later every modification, assemble update, or set of issue fixes, is difficult.[4]

1.6.9 Smoke Testing

This is a product testing strategy which surveys whether a framework bundle is conveyed and is looking after strength. "Construct Verification Testing" or "Certainty Testing" are different terms for smoke testing. To put it another way, we're verifying whether the primary highlights are working and if there are any masterpieces in the form we're trying.

This permits you to check whether the form is flawed, keeping you from sitting around and cash on additional testing. The smoke tests show that the construction is prepared for additional conventional testing. The essential objective of smoke testing is to uncover extreme issues from the beginning. Smoke tests are utilized to show that a framework is steady and meets the models. All information records, libraries, reusable modules, and designing parts important to achieve at least one item functionalities are remembered for a form.

Execution of this testing

The advancement group conveys the form in QA in this testing method. Analyzers perform experiments on the form subsequent to stepping through determinations of examination cases. The application's fundamental usefulness is tried by the QA group. The reason for this arrangement of experiments is to uncover development mistakes. On the off chance that these tests pass, the QA group will proceed onward to Functional Testing.

Need of smoke testing

Smoke testing is indispensable in programming advancement since it guarantees the framework's precision in the beginning phases. We can set aside time and cash thusly. As a result, smoke tests guarantee that the framework is in legitimate working request. Just when we've finished smoke testing will we start practical testing.

- Smoke testing will recognize the entirety of the form's masterpieces, and it will be done after the form has been delivered to QA. Smoke testing distinguishes most of blunders during the beginning stages of programming advancement.

•Smoke testing makes recognizing and adjusting serious imperfections a lot simpler.

- Smoke testing permits the QA group to distinguish application usefulness blemishes that may have created because of the new code.

- Smoke testing recognizes the most genuine mistakes.

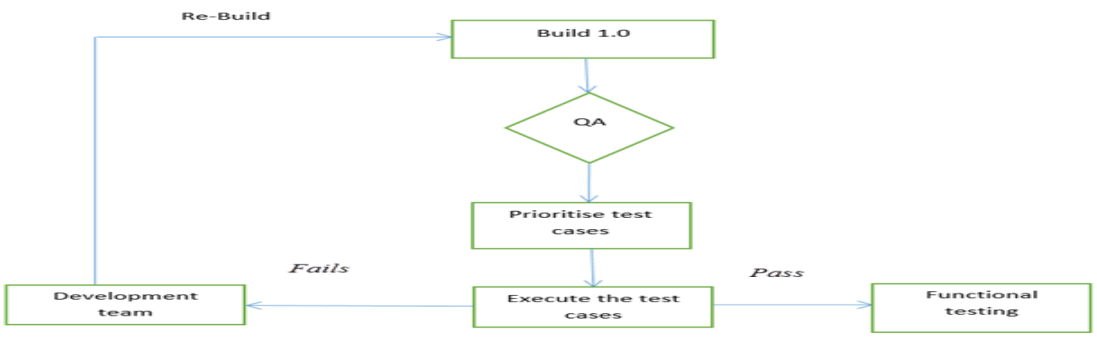


Figure1.4 Smoke test cycle

1.8 Test case writing

It is an arrangement of steps performed for guaranteeing that the specific usefulness or ability of our product program is working appropriately. Experiment implies an assortment of test strategies, information, preconditions, and postconditions made for a specific test situation all together for approving any necessity. The experiment contains determined conditions that an analyzer may use to count between expected outcomes and genuine outcomes all together for evaluating if the application addresses the client's issues.

1.7.1 Test Case Management Tools

These are mechanization arrangements which aid the organization and upkeep of Test Cases.

1. Documentation of Test Cases: You may use instruments to accelerate the development of Test Cases by utilizing formats.

2. Run the experiment and monitor the outcomes: The instruments might be utilized for running a test , and the yields might be helpfully recorded.

3. Automate Defect Tracking: The tests which don't pass are quickly associated with a bug tracker, thereafter it tends to be allotted to coding group and finished up on email alarms.

4. Traceability: Requirements, Test cases, and Test case execution are completely associated by the devices, additionally every experiment can measure up to the past one to assess the number of experiments are covered.

.

1.7.2 Benefits of Writing Test Cases

The fundamental objective of an experiment is to ensure that different functionalities of an application proceed as arranged. It helps analyzers in deciding whether the program is sans deformity and meets the prerequisites of end clients. Different benefits of experiments include:

- guaranteeing great test inclusion,
- assisting with upgrading programming quality, etc. Diminishes the expense of upkeep and programming support
- Help with guaranteeing that the program fulfills the necessities of the end client.
- Permits the analyzer to contemplate the tests and approach them from however many points of view as could reasonably be expected.
- Experiments are reusable later on since they might be referred to and executed by anyone.

1.7.3 Test Case Format

An ID, depiction, a lot of data, two or three vital exercises, and expected and genuine outcomes are the rule portions of an analysis.

- Experiment Name: An investigation should have a quite obvious name or title.

- Experiment Description: two or three words, the depiction should portray the analyzer what they'll test.

- Pre-Conditions: Include any speculations that apply to the test similarly as any preconditions that ought to be satisfied before the test can be run.

- Experiment Steps: The test steps should fuse the aggregate of the crucial data similarly as rules for running the test. The stages should be clear and compact, without notwithstanding huge information. Test Data: It's essential to pick a data arrangement that gives adequate incorporation. Pick a data combination that fuses extraordinary just as deplorable possible results.

- Expected Result: The ordinary outcomes depict what the analyzer should hope to see on account of the test strategies.

- Genuine Result: They portray how the application performed during the execution of investigations. Comments: Any additional information, for instance, screen catches, that the analyzer wishes to share might be added here.

This is the standard game plan used by analyzers when forming tests. Additional limits, for instance, test need, kind of analysis, and bug earnestness, can be added to these limits by analyzers.

Table 1.1: Test Case Example

Title	**Login with a valid username & password**
Precondition	User is already registered using valid credentials
Test Steps	1. Enter a valid username 2. Enter a valid password 3. Click on sign in
Expected Result	-User is logged in successfully -User is redirected to Home page Test Suite Login Test
Environment	Samsung Galaxy Note 10 - Android 10 – 4G Network

Actual Result	Same as expected
Status	Pass

1.9 Bug Report in Software Testing

In programming testing, a bug report is an exhaustive archive portraying the blemishes distinguished in the product program. A bug report includes everything about a bug, like the depiction, the date the bug was found, the character of the analyzer who found it, the name of the engineer who remedied it, etc. A bug report helps in the location of comparable issues later on, permitting them to be stayed away from.

The accompanying data ought to be remembered for the Problem Report when detailing the bug to the designer.

- Defect ID - The imperfection's special recognizing number.

- Defect Description-A nitty gritty portrayal of the Defect, remembering subtleties for the module where the Defect was found.

- Version - The program adaptation in which the defect was found.

- Steps - a point by point set of steps with screen captures that the engineer may use to imitate the defects.

Date Raised - Date when the deformity is raised

- Reference-where you give references to papers like as details, plan, design, or even pictures of the issue to help understand the shortcoming

- Detected By - The name/ID of the analyzer who revealed the blemish.

Status - Status of the deformity

Table 1.2: Defect Status

New (Ready to test)	The test case is not executed
Pass	The test case is executed and the actual result is the same as the expected result
Fail	The test case is executed and the actual result is different from the expected result
Blocked/Skipped	The test case can't be executed

- Fixed by - the engineer who settled it

- Date Closed - the date when the shortcoming was fixed

- Severity - the seriousness of the shortcoming's effect on the application

- Priority, which is attached to the earnestness of settling surrenders. As per the impact criticalness with which the issue ought to be redressed, the seriousness need may be High, Medium, or Low.

Table 1.3: Defect Report Example

Title	**Login -> Forgot password button isn't working**
Steps to reproduce	1. Click on Login 2. Click on Forgot password
Expected Result	The button can be clicked and user should be redirected to a page to enter his email
Actual result	Clicking on the button doesn't have any impact
Test Environment	Samsung Galaxy Note 10 - Android 10 – 4G Network
Priority	High
Type	Functional
Severity	High

1.8.1 Bug life cycle

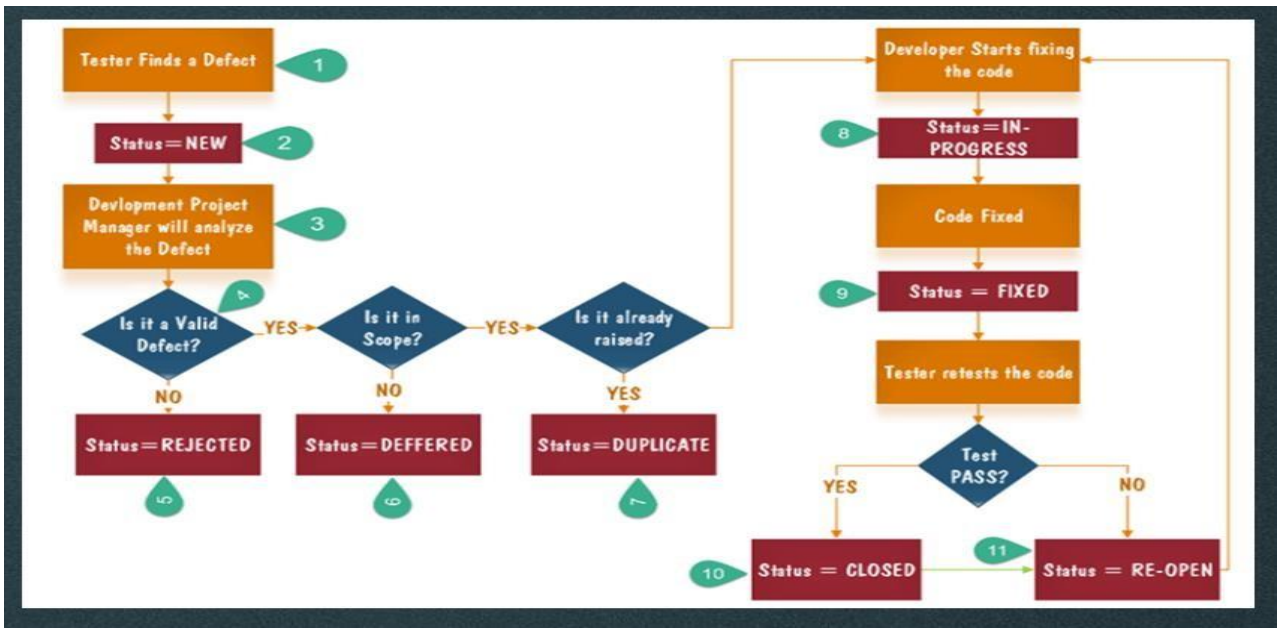


Figure1.5 Bug Life Cycle

CHAPTER 2

DATASOURCE

2.1 SQL

SQL is a relational database programming language for obtaining and managing data. The abbreviation for Structured Query Language (SQL) stands for Structured Query Language.

Structured Query Language (SQL) is a computer language for storing, manipulating, and retrieving data from a relational database.

SQL is the standard language for Relational Database Systems (RDBMS). All Relational Database Management Systems (RDMS) Server employ SQL as their main database language.

2.1.1 Applications of SQL

As recently said, SQL is perhaps the most broadly utilized information base question dialects.

- Provides admittance to information put away in social data set administration frameworks to clients.

- Allows clients to clarify information base.

- SQL modules, libraries, and pre-compilers might be utilized to insert in different dialects, permitting clients to characterize and adjust information in a data set. Clients may make and eliminate data sets and tables with this program.

- Provides clients with the capacity to make table inquiries, data set items, and activities.

- Users can change the limitations for sections, capacities, and articles.

2.1.2 SQL Process

In the event that we play out a Sql question towards any RDBMS, our product chooses the ideal way to deal with satisfy given solicitation and the SQL motor decides when to fathom it. There are a few parts to this activity. These are the components:

- Engines for enhancement

- Dispatcher of Query

- Search Engine (Classic)

- SQL Query Engine, for instance.

A customary question motor handles non-SQL inquiries, though a SQL inquiry motor handles consistent documents. The SQL Architecture is portrayed in the figure underneath.

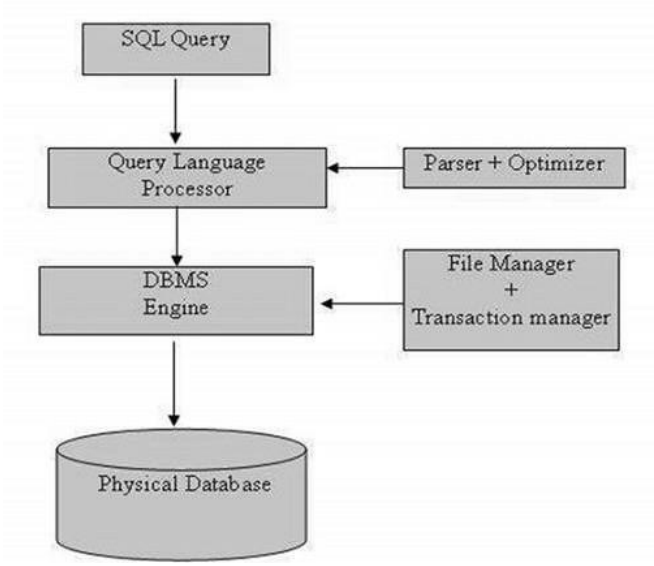


Figure2.1 SQL Architecture

2.1.2 RDBMS

"Social Database Management System" is an abbreviation for "Rdbms." A social data set administration framework, or RDBMS, is a data set administration framework made only for social data sets. RDBMS are along these lines a subset of DBMS.

It is one that utilizations lines and segments to store information in a coordinated manner. This makes finding and getting to specific qualities inside the data set a breeze. As its qualities inside each field are associated to each other, it is classified "social." Tables can likewise be connected to each other. In view of the social nature, inquiries might be directed on a few tables simultaneously.

2.2 XML

The shortened form for (XML) means "Extensible Markup Language." Despite Html components, which indicate how the information ought to be appeared, XML labels investigate the data and can be utilized for putting away and keeping up it as opposed to determining how it ought to be appeared. Sooner rather than later, XML would not wipe out HTML, however this will get to more data by joining a large number of HTML's astounding qualities.

Three fundamental qualities of XML make it valuable in a wide scope of frameworks and

arrangements:

- XML is extensible: We may plan our labels giving our own depiction and in any language, to accommodate our product.

XML moves information yet won't show it. For what reason does XML convey information yet not show it? You can save information in XML independent of how it very well may be shown.

- XML is a uninhibitedly available norm. The World Wide Web Consortium (W3C) created XML, that is free and open innovation.

2.2.1 Usage

XML could work behind at scenes to facilitate the age of HTML content for colossal sites, as indicated by a concise rundown of its employments.

Data might be sent among associations and organizations utilizing this. XML might be utilized to dump and reload data sets.

Information might be put away and coordinated utilizing XML, makes it simple to tweak specific social data set requirements.

XML in addition to templates might be effectively coupled to give practically any right result. Practically any sort of information might be sent utilizing a XML archive.

2.2.2 Markup Description

XML is a markup language for portraying of rules for order perceptions in a manner that is both physical and computerized comprehensible. Markup is data that is attached to something like an archive to improve its significance surely, for example, recognizing the components and their connections. It is a bunch of images that might be placed into a record's book to recognize and mark its various bits.

2.3 JSON

JSON, or JavaScript Object Notation, is a book based open norm for moving comprehensible information. Software engineers know about JSON shows and other programming dialects.

- JSON design was made by Douglas Crockford

- It was made to trade information in a comprehensible arrangement.

- The JavaScript programming language has been extended.

- The .json record expansion is utilized.

- JSON is a kind of information design. Application/json is the Internet Media Type.

- The Uniform Type Identifier (UTI) is available to general society. json is a sort of information.

2.3.1 Uses of JSON

- It's used in the improvement of uses which are java based, like augmentations of program and website pages.

- It is a serialization or transmission design for information which is organized over an association of organizations.

- This is for the most part utilized for send data from a worker to site applications.

- JSON is the arrangement utilized by online administrations and APIs to uncover information which isn't private.

- This is viable by the present programming dialects.

2.2.1 JSON Characteristics

- It is basic in comprehension and composing.

- This is a tradable arrangement which depends on text and that is not difficult to utilize.

- JSON is language rationalist.

CHAPTER 3

VBSCRIPT

Visual Basic Scripting, or VBScript, is a subset of Visual Basic for Applications (VBA). This is a Microsoft item that might be found not simply in Microsoft items like MS Project and MS Office, and can be utilized in various projects like AUTO CAD.

3.1 VBScript highlights

- It's a basic prearranging language with a lightning-quick processor.

- VBScript is case-unfeeling generally. It has an exceptionally straightforward grammar that is not difficult to get and use.

•VBScript isn't an Object-Oriented Programming language like C++ or Java, yet rather an item based prearranging language.

- To access the parts of the climate wherein it is running, it utilizes the Component Object Model (COM).

- Only in a Host Environment, like Internet Explorer (IE), Internet Information Services (IIS), or Windows Scripting Host, can VBScript be effectively run (WSH)

3.2 VBscript – Version History and Uses

Microsoft delivered the principal form of VBScript in 1996, and it was variant one. Latest stable form of VBScript is 5.8, which is incorporated with Internet Explorer 8 and Windows 7. Where VBScript might be utilized are various and are not restricted to the ones recorded underneath.

- Quick Test Professional, truncated as QTP, is a famous robotization testing application that utilizations VBScript as a prearranging language.

- Windows Scripting Host, which is for the most part utilized for computerizing the Windows Desktop by Windows System Administrators.

- Active Server Pages is a prearranging climate which has worker that utilizes VBScript or Java Script to make dynamic pages.

- In Microsoft Internet Explorer, VBScript is used for customer side prearranging.

•VBScript is utilized to execute Microsoft Outlook Forms, though VBA is utilized for application-level programming.

Disservices

- Only Internet Explorer programs support VBscript. VBScript isn't upheld by different programs, like Chrome and Firefox. Therefore, JavaScript overshadows VBScript.

- VBScript's order line usefulness is restricted.

•Debugging is extreme because of the absence of a default improvement climate.

VBScript use

The most recent adaptation of VBScript is 5.8, likewise with the most recent presentation of the.NET structure, Microsoft has selected to keep on supporting it for web improvement inside ASP.NET. Subsequently, no new forms of the VBScript motor will be delivered, anyway the Microsoft supporting Engineering Team will deal with any imperfection fixes and security issues. The VBScript motor, then again, would be incorporated naturally in all Microsoft Windows and IIS establishments.

CHAPTER 4
UFT AUTOMATION

Bound together Functional Testing, or UFT, is a computerized utilitarian testing instrument that helps analyzers in executing robotized tests to discover any slip-ups, blames, and spaces inside application under test that are not steady with the normal outcomes. Mercury Interactive made it, which was ultimately purchased by HP and is presently Microfocus. QTP represents Quick Test Professional, while UFT represents Unified Functional Testing.

UFT, initially Quick Test Professional (QTP), is a piece of programming that computerizes utilitarian and relapse testing for programming applications and conditions.

It has a graphical UI and supports catchphrase and prearranging interfaces. It indicates a test technique and controls the articles and controls of the application under test utilizing the Visual Basic Scripting Edition (VBScript) prearranging language. From a solitary reassure, designers may test each of the three layers of a program's activities: the interface, the assistance layer, and the data set layer.

Speedy Test Professional was the underlying name for UFT, which was made by Mercury Interactive. Hewlett-Packard (HP) in the long run purchased Mercury Interactive in 2006. Up until 2016, when the HP Software Division was offered to Micro Focus, UFT 11.5 coordinated HP Quick Test Professional and HP Service Test into a solitary programming bundle that was available through the HP Software Division.

4.1 DESCRIPTION

This is a kind of programming that mechanizes the trial of an assortment of programming applications and settings. It utilizes a UI, like a local GUI or a web interface, to do utilitarian and relapse testing. It works by perceiving objects in an application's UI or a page and performs determined exercises; it might likewise record object information like property and id of controller. To portray the test technique and

alter the articles and controls of the application under test, This product utilizes the VBScript prearranging language. Clients may have to change the hidden to execute more intricate activities.[7]

Exemption dealing with

Miniature Focus UFT handles exemptions through recuperation situations, with the reason for proceeding to execute tests regardless of whether there is an unforeseen disappointment. Since UFT interfaces further into framework memory of the projects being tried, different occasions may make HPE Unified Functional Testing cut short and be unrepairable.

Information driven testing

Zero in on the little subtleties Data-driven testing is empowered at UFT. Data could be sent out to a dominate sheet, for instance, and reused elsewhere. This is completed utilizing a Microsoft Excel worksheet which might be moved toward utilizing UFT. The Global information sheet and the Action (neighborhood) information sheet are the two sorts of information tables utilized by UFT. The test stages can take information from these information tables and drive variable information into the application under test, guaranteeing that an expected result is accomplished.

Mechanizing custom and complex UI objects

Zeroing in on the little subtleties Customized UI things and other confounded things might be missed by UFT. These things can be named virtual items or knowledge objects by clients. Virtual articles are not really upheld by UFT for simple account or recording in low-level mode.

Extensibility

This might be tweaked with additional items having a wide scope of improvement conditions which aren't conveyed outside the application. Web, .NET, Java, and Delphi are completely upheld by UFT add-ins. HP Functional Testing programming incorporates HP Quick Test Professional and the HP Quick Test

Professional add-ins. UI Design

It gives two perspectives on a test script, just as choices to modify it: Keyword View and Expert View. These perspectives empower UFT to work as a test's Integrated Development Environment (IDE), and UFT contains a few customary IDE abilities, for example, breakpoints, which permit you to end a test at specific focuses.

Clients may use Keyword View to construct or check test methodology in a particular, plain way. Each line shows a stage that can be changed. Any of the accompanying segments can be remembered for the Keyword View: Item, Operation, Value, Assignment, Comment, and Documentation. UFT shows a relating line of content for each stage in the Keyword View dependent on the worth of line and section.

Whenever, clients can add, erase, or modify steps.

Clients may likewise examine credits for objects like designated spots, yield esteems, and activities in Keyword View, just as use restrictive and circle articulations and supplement breakpoints to help troubleshoot a test.

It permits individuals to view or change the source code of a test utilizing VBScript. Analyzers can modify all test ventures aside from the root Global activity, and changes are adjusted with the watchword see, which is intended for more experienced clients.

The prearranging language utilized by Micro Focus UFT is VBScript. Classes are upheld by VBScript, anyway polymorphism and legacy are not. VBScript loses the capacity to use some Visual Basic watchwords, just as an incorporated debugger, an occasion controller, and a structures proofreader, when contrasted with Visual Basic for Applications (VBA). Despite the fact that HP has incorporated a debugger, it is confined in contrast with testing instruments that incorporate a full-highlighted IDE, for example, those included with other programming dialects.

4.2 Drawbacks

Miniature Focus UFT is a Windows-based application. It depends on fairly antiquated Windows-just innovation like ActiveX and VBScript, neither of which is an article situated language. This product unfit

to test with all program variants and sorts. It doesn't, for instance, support Opera.

Albeit far off execution is as yet accessible with HPE Unified Functional Testing running on a different framework, the Test Execution motor is combined with the GUI Test Code improvement IDE, accordingly it is extremely unlikely to run the tests freely of UFT.

Due to the high permit charges, the innovation is every now and again just used by a little gathering of analyzers inside an organization. This backings a compartmentalized way to deal with QA/testing, where test is done independently from the business and advancement groups, as opposed to a cooperative methodology where analyzers work together intimately with the engineers and business group.

CHAPTER 5

DESCRIPTION OF WORK CARRIED OUT

5.1 Writing test cases for hotel booking website

Test Scenario

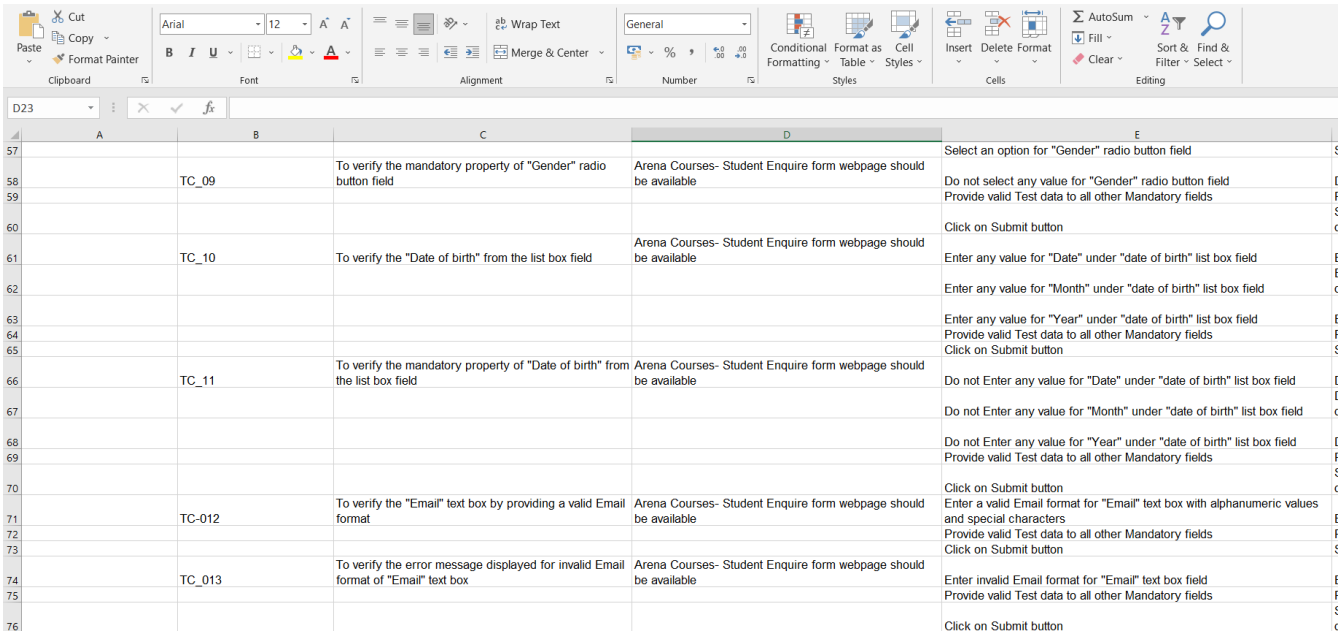
Module	Scenario ID	Scenario Name	Scenario Description	Requirement id
Arena Courses Website- Student Enquiry form	TS_01	Find Course- Valid	To check whether the user is able to find course if he/she provides valid details	RID_01
	TS_02	Find Course- Invalid	To check whether the user is able to find course if he/she provides invalid details	
	TS_03	Find Course- Null Values	To check whether the user is able to find course by not providing any details	RID_02
	TS_04	Find Course- Alternate Flow	To check whether the user is able to find course if he/she selects the cancel button	RID_03
	TS_05	Fields Validation	To Validate all the possible fields available for Submit functionality	

Test Cases

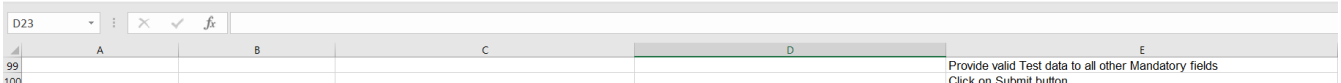
D23

	A	B	C	D	E
2	TS_01	TC_01	To verify the Submit Functionality by providing the valid inputs	Arena Courses- Student Enquire form webpage should be available	Step1. Enter a valid value for "Name" textbox field
3					Step2. Select "Gender" option for radio button field
4					Step3. Select the "Date of birth" from the list box in DD-Month-Year format
5					Step4. Enter a valid value for "Email ID" textbox field
6					Step5. Enter a valid value for "phone" textbox field
7					Step6. Enter a valid value for "city" textbox field
8					Step7. Enter a valid value for "Pincode" textbox field
9					Step8. Select a valid value for choosing a course in "Interested in" list box field
10					Step8. Enter a valid value for "Comments" textbox field
11					Step9. Click on the submit button
12					
13					
14	TS_01	TC_02	To check the **submit** functionality for the courses by filling the Invalid "Students enquiry form"	Arena Courses- Student Enquire form webpage should be available	**Enter an Invalid value for "Name" textbox field**
15					Select "Gender" option for radio button field
16					Select the "Date of birth" from the list box in DD-Month-Year format
17					Enter an Invalid value for "Email ID" textbox field
18					Enter an Invalid value for "phone" textbox field
19					Enter an Invalid value for "city" textbox field
20					Enter an Invalid value for "Pincode" textbox field
21					Select a valid value for choosing a course in "Inserted in" list box field
22					Enter an Invalid value for "Comments" textbox field

TS_02	TC_03	To Check the Submit functionality for the courses by filling null inputs in "Students Enquire form"	Arena Courses- Student Enquire form webpage should be available	Do not Enter a valid value for "Name" textbox field
				Do not Select "Gender" option for radio button field
				Do not Select the "Date of birth" from the list box in DD-Month-Year format
				Do not Enter a valid value for "Email ID" textbox field
				Do not Enter a valid value for "phone" textbox field
				Do not Enter a valid value for "city" textbox field
				Do not Enter a valid value for "Pincode" textbox field
				Do not Select a valid value for choosing a course in "Interested in" list box field
				Do not Enter a valid value for "Comments" textbox field
				Click on Submit button
TS_03	TC_04	To verify "Name" field by entering valid Name having only alphabets values of maximum 30 characters	Arena Courses- Student Enquire form webpage should be available	Enter a valid name into "Name" textbox field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_05	To verify "Name" field by entering non-alphabetic values	Arena Courses- Student Enquire form webpage should be available	Enter an Invalid name value for "Name" textbox field with non-alphabetic values
				Provide valid Test data to all other Mandatory fields

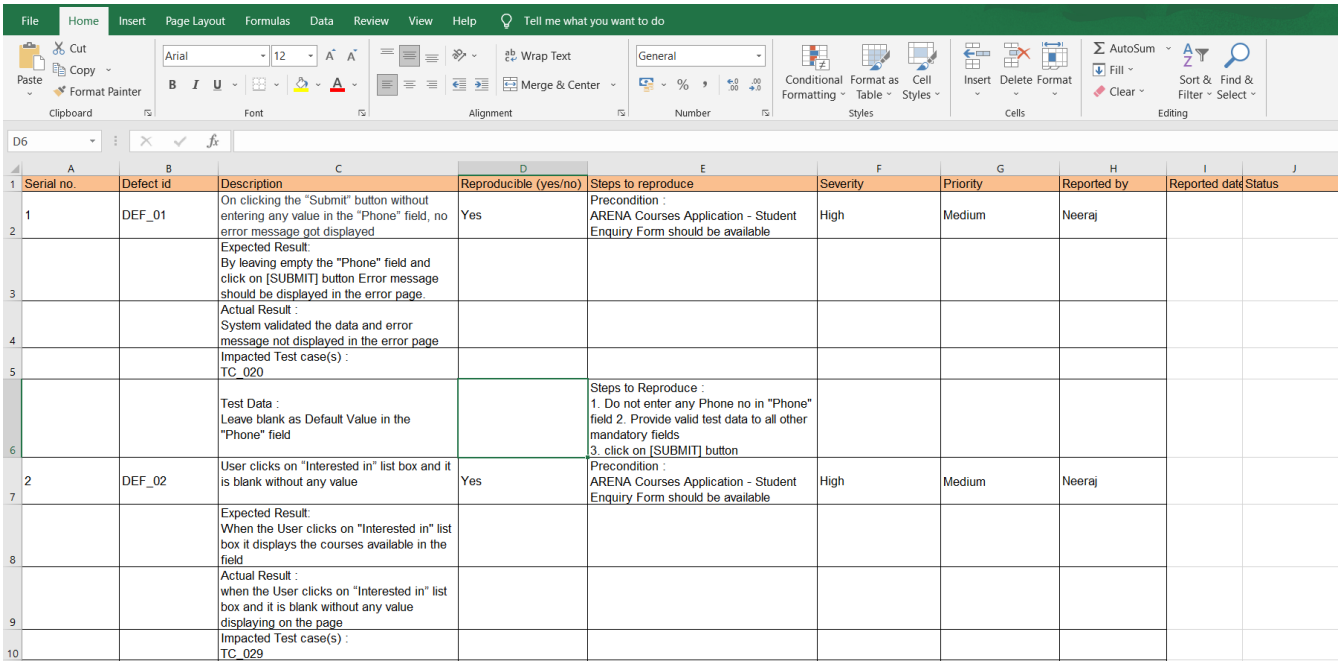


				Select an option for "Gender" radio button field
	TC_09	To verify the mandatory property of "Gender" radio button field	Arena Courses- Student Enquire form webpage should be available	Do not select any value for "Gender" radio button field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_10	To verify the "Date of birth" from the list box field	Arena Courses- Student Enquire form webpage should be available	Enter any value for "Date" under "date of birth" list box field
				Enter any value for "Month" under "date of birth" list box field
				Enter any value for "Year" under "date of birth" list box field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_11	To verify the mandatory property of "Date of birth" from the list box field	Arena Courses- Student Enquire form webpage should be available	Do not Enter any value for "Date" under "date of birth" list box field
				Do not Enter any value for "Month" under "date of birth" list box field
				Do not Enter any value for "Year" under "date of birth" list box field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC-012	To verify the "Email" text box by providing a valid Email format	Arena Courses- Student Enquire form webpage should be available	Enter a valid Email format for "Email" text box with alphanumeric values and special characters
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_013	To verify the error message displayed for invalid Email format of "Email" text box	Arena Courses- Student Enquire form webpage should be available	Enter invalid Email format for "Email" text box field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button

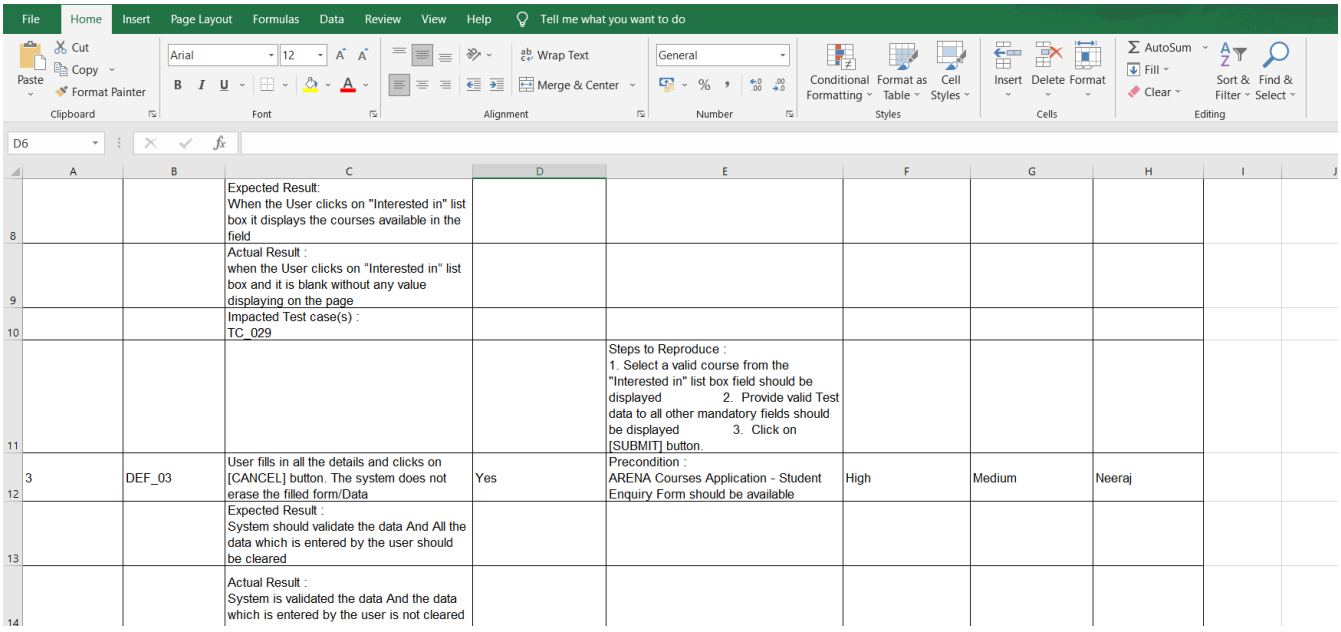


				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_22	To verify the error message displayed by providing invalid input of less than 6 digits for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter an invalid value for "Pincode" text box with less than 6 digits
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_23	To verify the error message displayed by providing invalid input of greater than 6 digits for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter an invalid value for "Pincode" text box with greater than 6 digits
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_24	To verify the error message displayed by providing other than numeric input for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter invalid input with non-numeric values for "Pincode "text box
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_025	To verify the database values for "interested in" list box field	Arena Courses- Student Enquire form webpage should be available	Click on the "interested in " list box field
				Check for the values displayed for "interested in" list box field
				select a valid value for choosing a course in "Interested in" list box field
	TC_026	To validate the "comment" text box by providing valid input having alphanumeric values of Maximum 500 Characters.	Arena Courses- Student Enquire form webpage should be available	Enter a valid input for "comments" text box
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_27	To validate the error message displayed by giving input greater than 500 characters in length for "comments" text box	Arena Courses- Student Enquire form webpage should be available	Enter an Invalid value for "Comments" textbox field with more than 500 characters

Defect Report

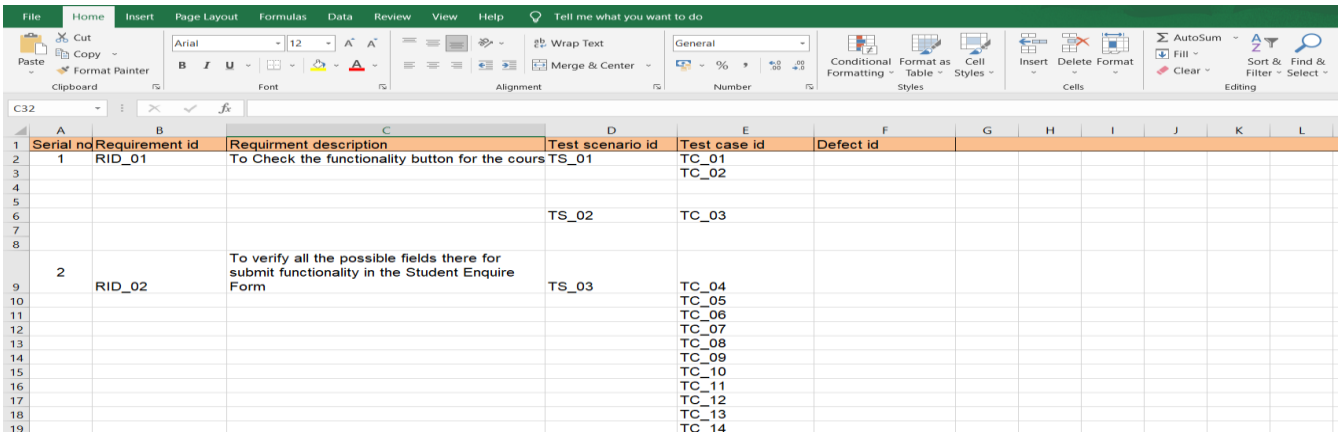


Serial no.	Defect id	Description	Reproducible (yes/no)	Steps to reproduce	Severity	Priority	Reported by	Reported date	Status
1	DEF_01	On clicking the "Submit" button without entering any value in the "Phone" field, no error message got displayed	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj		
		Expected Result: By leaving empty the "Phone" field and click on [SUBMIT] button Error message should be displayed in the error page.							
		Actual Result : System validated the data and error message not displayed in the error page							
		Impacted Test case(s) : TC_020							
		Test Data : Leave blank as Default Value in the "Phone" field		Steps to Reproduce : 1. Do not enter any Phone no in "Phone" field 2. Provide valid test data to all other mandatory fields 3. click on [SUBMIT] button					
2	DEF_02	User clicks on "Interested in" list box and it is blank without any value	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj		
		Expected Result: When the User clicks on "Interested in" list box it displays the courses available in the field							
		Actual Result : when the User clicks on "Interested in" list box and it is blank without any value displaying on the page							
		Impacted Test case(s) : TC_029							



		Expected Result: When the User clicks on "Interested in" list box it displays the courses available in the field					
		Actual Result : when the User clicks on "Interested in" list box and it is blank without any value displaying on the page					
		Impacted Test case(s) : TC_029					
				Steps to Reproduce : 1. Select a valid course from the "Interested in" list box field should be displayed 2. Provide valid Test data to all other mandatory fields should be displayed 3. Click on [SUBMIT] button.			
3	DEF_03	User fills in all the details and clicks on [CANCEL] button. The system does not erase the filled form/Data	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj
		Expected Result : System should validate the data And All the data which is entered by the user should be cleared					
		Actual Result : System is validated the data And the data which is entered by the user is not cleared					

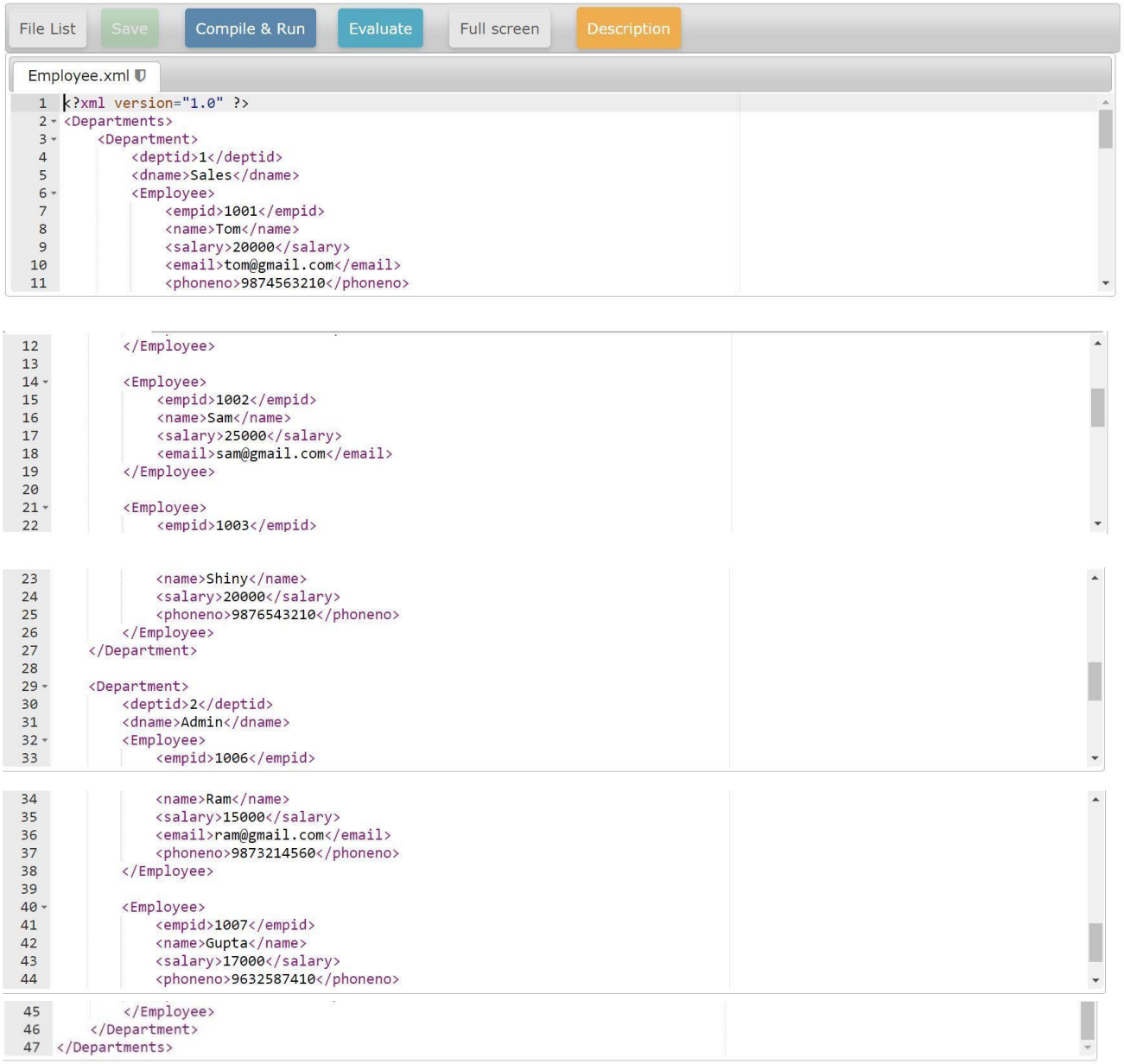
RTM (Requirement Traceability Matrix)



Serial no	Requirement id	Requirment description	Test scenario id	Test case id	Defect id
1	RID_01	To Check the functionality button for the cours	TS_01	TC_01	
				TC_02	
			TS_02	TC_03	
2	RID_02	To verify all the possible fields there for submit functionality in the Student Enquire Form	TS_03	TC_04	
				TC_05	
				TC_06	
				TC_07	
				TC_08	
				TC_09	
				TC_10	
				TC_11	
				TC_12	
				TC_13	
				TC_14	

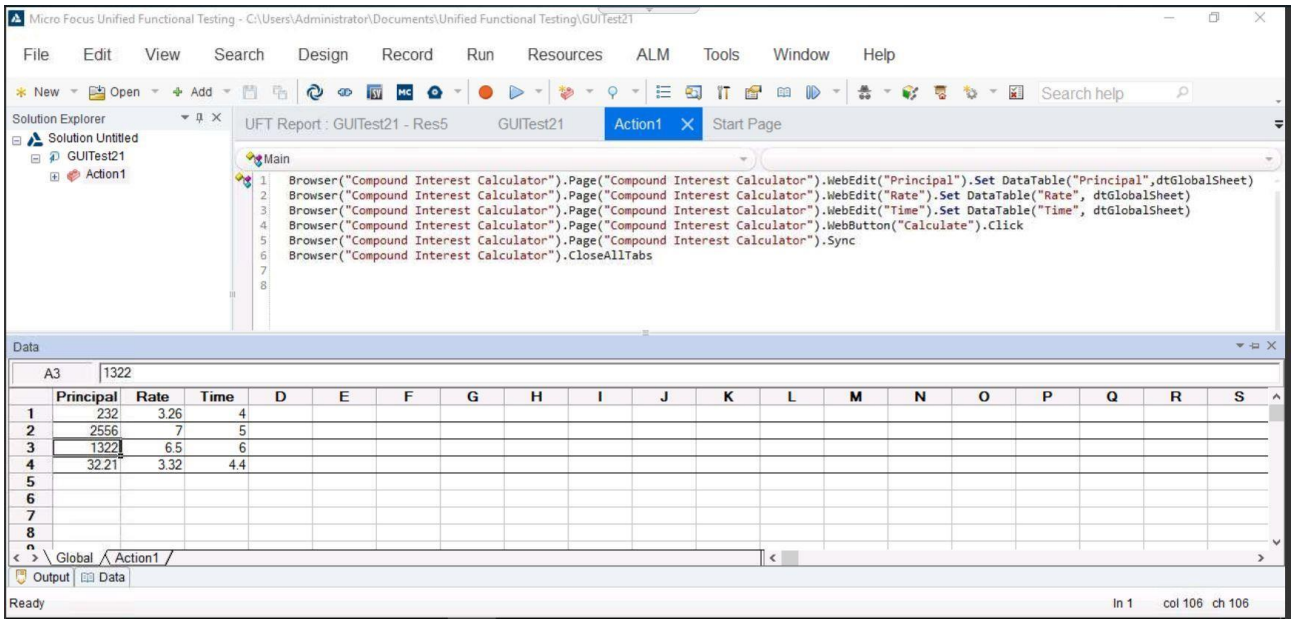
5.2 Creating a well-formed document for the given scenario using XML

Deptid	Dname	Empid	Name	salary	email	phoneno
1	Sales	1001	Tom	20000	tom@gmail.com	9874563210
1	Sales	1002	Sam	25000	sam@gmail.com	
1	Sales	1003	Shiny	20000		9876543210
2	Admin	1006	Ram	15000	ram@gmail.com	9873214560
2	Admin	1007	Gupta	17000		9632587410

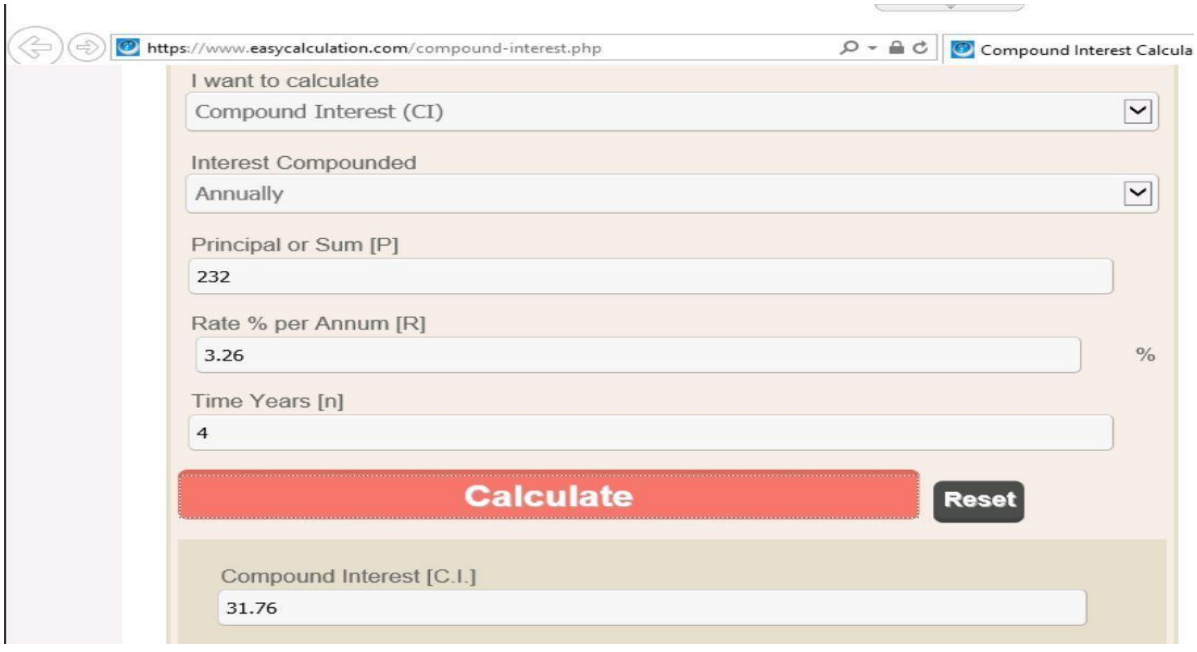


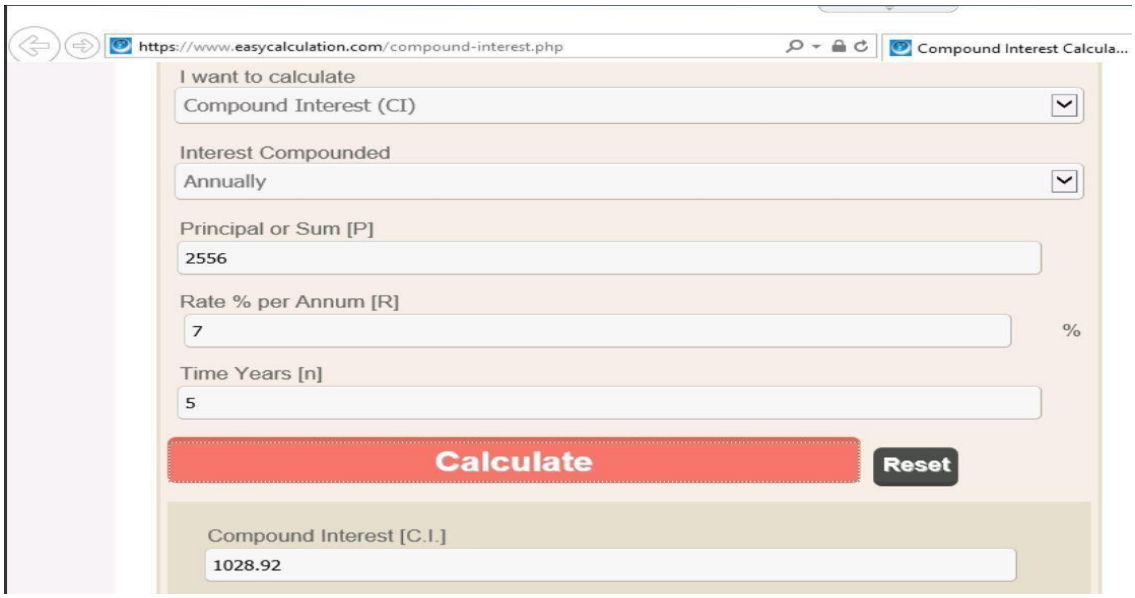
5.3 To check the functionality of calculate button for website http://easycalculation.com/compound-interest.php for different inputs using UFT Automation.

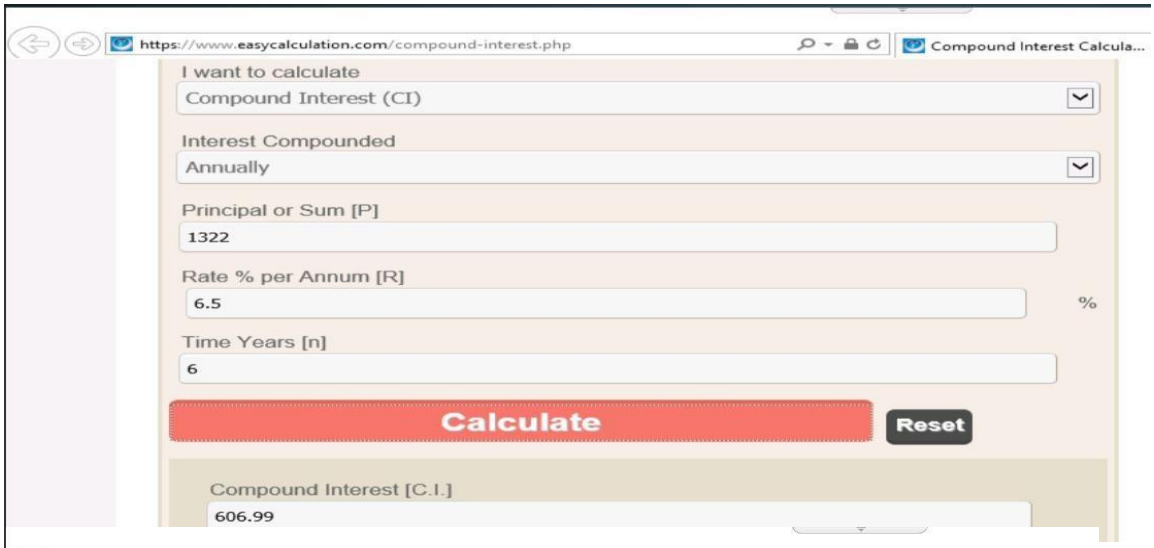
Script

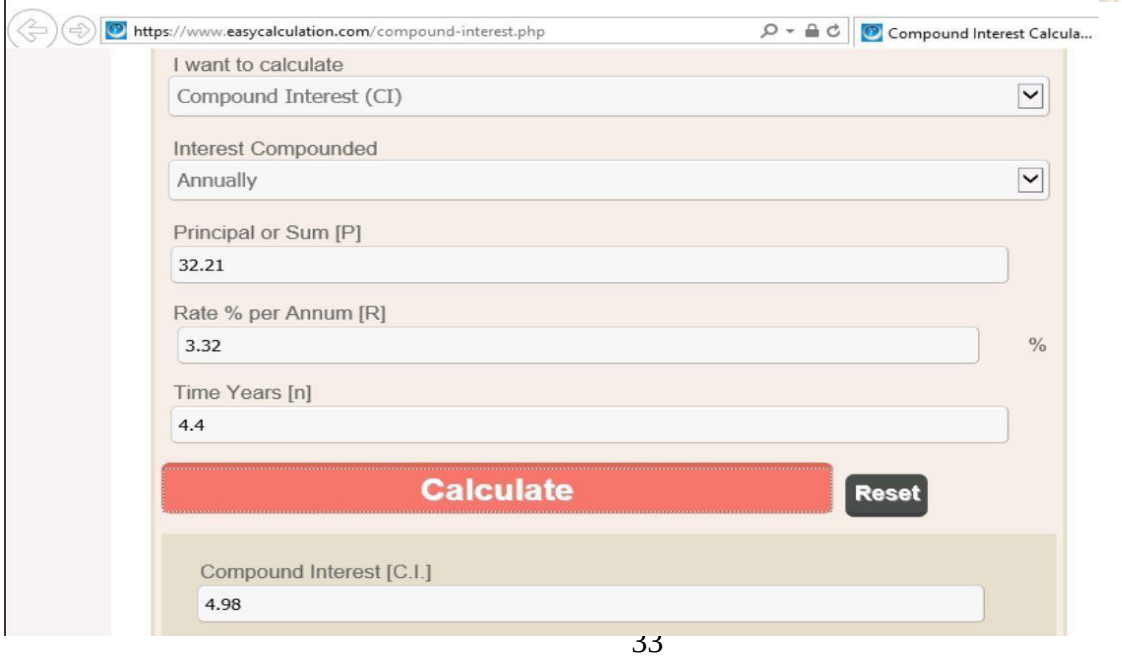


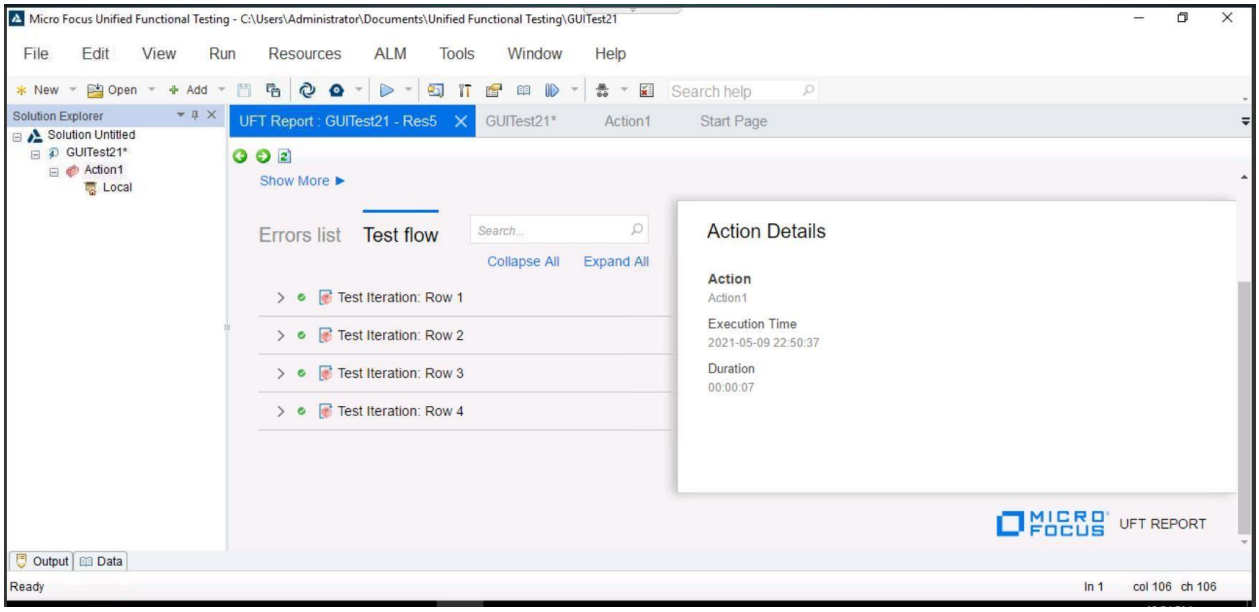
Outputs

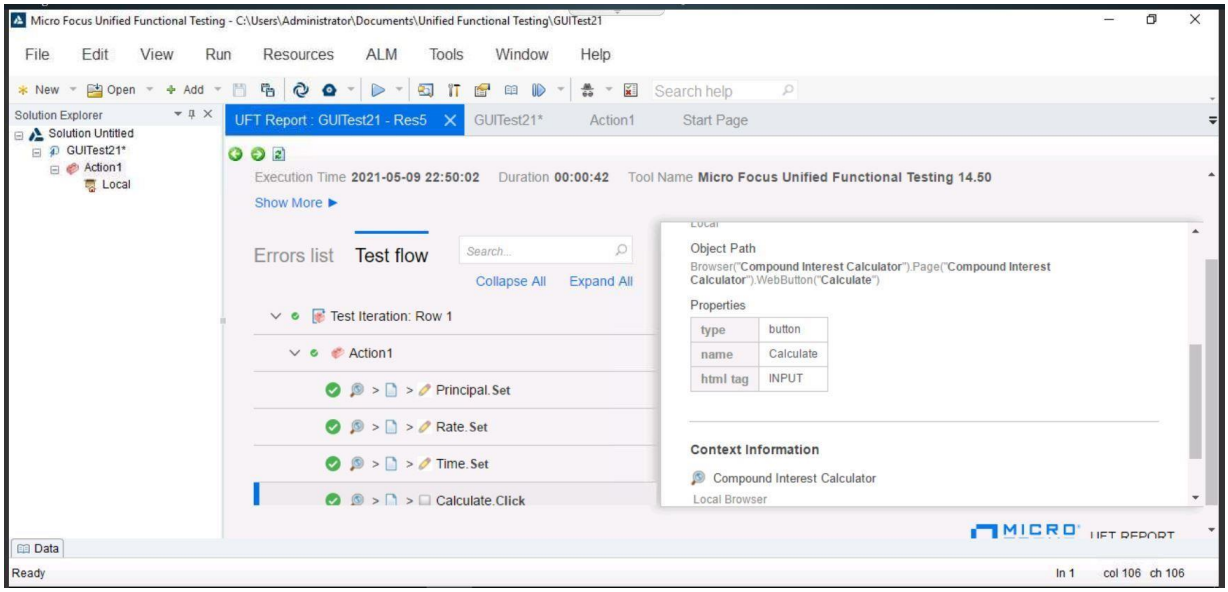



https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
2556
Rate % per Annum [R]
7
%
Time Years [n]
5
Calculate
Reset
Compound Interest [C.I.]
1028.92


https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
1322
Rate % per Annum [R]
6.5
%
Time Years [n]
6
Calculate
Reset
Compound Interest [C.I.]
606.99


https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
32.21
Rate % per Annum [R]
3.32
%
Time Years [n]
4.4
Calculate
Reset
Compound Interest [C.I.]
4.98


Micro Focus Unified Functional Testing - C:\Users\Administrator\Documents\Unified Functional Testing\GUITest21
File Edit View Run Resources ALM Tools Window Help
New Open Add Search help
Solution Explorer
Solution Untitled
GUITest21*
Action1
Local
UFT Report : GUITest21 - Res5
GUITest21*
Action1
Start Page
Show More ►
Errors list
Test flow
Search...
Collapse All
Expand All
Test Iteration: Row 1
Test Iteration: Row 2
Test Iteration: Row 3
Test Iteration: Row 4
Action Details
Action
Action1
Execution Time
2021-05-09 22:50:37
Duration
00:00:07
MICRO FOCUS UFT REPORT
Output
Data
Ready
ln 1 col 106 ch 106


Micro Focus Unified Functional Testing - C:\Users\Administrator\Documents\Unified Functional Testing\GUITest21
File Edit View Run Resources ALM Tools Window Help
New Open Add Search help
Solution Explorer
Solution Untitled
GUITest21*
Action1
Local
UFT Report : GUITest21 - Res5
GUITest21*
Action1
Start Page
Execution Time 2021-05-09 22:50:02 Duration 00:00:42 Tool Name Micro Focus Unified Functional Testing 14.50
Show More ►
Errors list
Test flow
Search...
Collapse All
Expand All
Test Iteration: Row 1
Action1
Principal.Set
Rate.Set
Time.Set
Calculate.Click
Object Path
Browser("Compound Interest Calculator").Page("Compound Interest Calculator").WebButton("Calculate")
Properties
type button
name Calculate
html tag INPUT
Context Information
Compound Interest Calculator
Local Browser
Data
Ready
ln 1 col 106 ch 106

CHAPTER 6
CONCLUSION

Miniature Focus' UFT One is device that utilizes robotize tests to discover issues of a test application. Brought together Functional Testing (UFT) is an abbreviation for "Bound together Functional Testing." It was recently known as QTP (Quick Test Professional).

Practical, relapse, and administration testing are the most well-known utilizations for UFT. We may utilize UFT to mechanize client exercises on a site or worker PC program, and afterward test and identify abandons for numerous clients, different informational collections, various Windows working frameworks, or potentially various gadgets utilizing similar tasks. When contrasted with manual testing, robotization by means of UFT may save a great deal of time and cash if all around planned and done. Today, UFT is quite possibly the most ordinarily used business robotized testing arrangements accessible. It's notable for its effortlessness of utilization and merchant support, just as a major local area of mechanization specialists. Accordingly, qualified UFT experts have consistently been sought after

In this manner, utilizing programming testing methods experiments were composed for the specific application. Also, with the assistance of UFT computerization instrument we're ready to test the usefulness of the site effectively.

REFERENCES

[1] Prasad, Dr. K.V.K.K. (2008) ISTQB Certification Study Guide, Wiley, ISBN 978-81-7722-711-6, p.

[2] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/software_testing_dictionary/functional_testing.htm

[3] Krishna Rungta - Founder & CEO, Feb. 2013. Accessed on: Nov. 2, 2017. [Online]. Available: https://www.guru99.com/functional-testing.html

[4] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/software_testing/software_testing_types.htm

[5] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online]. Available: https://www.tutorialspoint.com/sql/index.htm

[6] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/vbscript/index.htm

[7] "HP-UFT 11.50 (Unified Functional Testing)". Retrieved August 10, 2018. Available: selftechy.com.

[8] "QTP with Descriptive programming". Retrieved January 27,2011. Available: Slideshare.net.

FinalReport_171343_Neeraj_Gupta.pdf

by

Submission date: 24-May-2021 06:23PM (UTC+0530)
Submission ID: 1593119162
File name: FinalReport_171343_Neeraj_Gupta.pdf (2.07M)
Word count: 8901
Character count: 48542

AUTOMATION TESTING -HC FACETS WITH UFT

Project report submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

By

Neeraj Gupta (171343)

UNDER THE GUIDANCE OF

Cognizant

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May, 2021

TABLE OF CONTENTS

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled **"Automation Testing-HC Facets with UFT"** submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of my work carried out under the supervision of Cognizant. I have not submitted this work elsewhere for any other degree or diploma.

Neeraj Gupta

171343

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Cognizant

Date: 24/05/2021

Head of the Department/Project Coordinator

Project Report Undertaking

I **Mr.** /Ms. Neeraj Gupta, Roll No. 171343, Branch Computer Science Engineering is doing my internship with Cognizant from 27 February 2021 to 11 June 2021.

As per the procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Name: Neeraj Gupta

Roll No.: 171343

Date: 24/05/2021

ACKNOWLEDGEMENT

My success of completion of my project required guidance from many individuals and assistance from many people and I am extremely privileged to have got this all along the completion of this project.

I take this opportunity to express my gratitude to our supervisor Dr. Himanshu Jindal, for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his constant encouragement and advice throughout our project.

The in-house facilities provided by the department throughout the project are also equally acknowledgeable.

LIST OF ACRONYMS AND ABBREVIATIONS

UFT	Unified Functional Testing
QTP	Quick Test Professional
SQL	Structured Query Language
VBScript	Visual Basic Script
XML	Extensible Markup Language
JSON	Java Script Object Notation
IE	Internet Explorer
SDLC	Software Development Life cycle
QA	Quality Assurance

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Micro Focus UFT One is instrument that utilizes computerize tests to discover issues of a test application. Unified Functional Testing (UFT) is an abbreviation for "Bound together Functional Testing." It was recently known as QTP (Quick Test Professional).

Utilitarian, relapse, and administration testing are the most widely recognized utilizations for UFT. You may utilize UFT to robotize client exercises on a site or worker PC program, and afterward test and recognize surrenders for numerous clients, different informational collections, various Windows working frameworks, and additionally various gadgets utilizing similar activities. When contrasted with manual testing, computerization through UFT may save a great deal of time and cash if very much planned and done.

Today, UFT is perhaps the most generally used business mechanized testing arrangements accessible. It's notable for its effortlessness of utilization and merchant support, just as a major local area of mechanization specialists. Accordingly, qualified UFT experts have consistently been popular.

CHAPTER 1
FUNCTIONAL TESTING

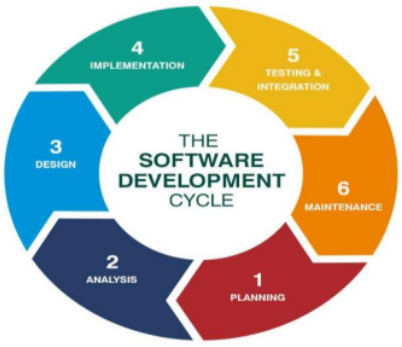
Software testing is a strategy for assessing program quality and bringing down the danger of programming mistake while being used. Most of people have experienced programming that doesn't proceed as arranged. Programming that doesn't perform as expected can cause an assortment of issues, including monetary misfortune, lost time, and harm to an organization's image. Passing or injury

Test Execution is a piece of programming testing. Programming testing is a multi-step measure that includes an assortment of assignments. One of these activities is execution. Programming testing is an innovation for finding if the program matches determined rules and affirming that it is sans blunder. It utilizes mechanization or manual testing for getting to numerous characteristics of interest by dealing with programming segments. Rather than genuine prerequisites, programming's testing will likely discover missteps, holes, and legitimate mistakes.

1.1 Software Development Life Cycle (SDLC)

The Life Cycle for Software Development (SDLC) is an interaction for programming plan, improvement and testing. The SDLC offers top notch items that fulfill customer necessities and are finished on schedule and on time. The SDLC is a strategy for a product project inside a product firm. It is a finished strategy to show how explicit programming might be assembled, overseen, refreshed and improved. The existence cycle is a method for improving the nature of programming and the improvement process.

The numerous means of a common SDLC are portrayed graphically in the outline underneath.


4
IMPLEMENTATION
5
TESTING &
INTEGRATION
3
DESIGN
THE
SOFTWARE
DEVELOPMENT
CYCLE
6
MAINTENANCE
2
ANALYSIS
1
PLANNING

Synotive

Figure1.1 Software Development Life Cycle

A typical Software Development Life Cycle consists of the following stages –

1 Planning

Examination of necessities is the most huge and principal phase of the SDLC. It is finished by the top individuals from the group with client criticism, outreach groups, market studies and experts from the business. These information are then used to characterize the center venture plan and complete item research drives in the financial, business and innovative fields.

2 Defining Requirements

The requests of the item should be fittingly determined, recorded and approved by the client or venture supervisors following the necessity examination. This is accomplished by utilizing a SRS (Software Requirement Specifications) record that incorporates all venture results to be set up and created through the entire task life cycle.

3 Designing the Product Architecture

For item designers needing to assemble the best engineering for another item, SRS is an asset. In a DDS – Specification of the plan report, various options in contrast to the idea recommended are typically given and systematized by the standards set out in the SRS.

All key partners evaluate this DDS and select an item plan procedure dependent on a few highlights, for example, hazard the executives, item steadiness, plan adaptability, spending plan and timescales.

4 Building or Developing the Product

At this period of the SDLC the real advancement of the item starts. The programming code is created as per DDS at this stage. At the point when the plan is done precisely and underlying, code age might be done quickly and without any problem.

5 Testing the Product

Since testing in present day SDLC plans is principally consolidated in all periods of SDLC, this stage is frequently an all-stage subset. This stage by and by concerns simply the testing period of the item, during which issues are accounted for, observed, fixed and checked until the item satisfies the SRS quality necessities.

6. Deployment in the Market and Maintenance

When the item is totally tried and prepared to send, it is authoritatively distributed on the applicable market. Contingent upon the plan of action of the firm, item carry out frequently happens in stages. Contingent upon input, the item would then be able to be delivered as or with recommended upgrades in the market bunch. It is kept up for the current client base after the items have been delivered to the market.

1.2 Software Testing Objectives

Figure1.2 Testing Objectives

To evaluate the work products

Before the engineer gets the work things for improvement, like the Requirements determination, Designs, and User Stories, they ought to be checked. Perceiving any vulnerability or clashing prerequisites now saves a ton of time during advancement and testing. Before the code incorporates/is prepared for testing, it goes through static examination (audits, walk-throughs, assessments, etc). Confirmation is the term for this kind of assessment. During the advancement period of any work item, it is the way toward investigating the item.

To verify requirements:

This objective exhibits that perhaps the main parts of testing is to fulfill the needs of the customer.

Analyzers look at the item and guarantee that the entirety of the specified guidelines are met. Planning each experiments, regardless of testing approach, ensures that each performed experiment is practically confirmed. A necessity detectability network (RTM) ought to likewise be made by the analyzer

to ensure that all experiments are planned to prerequisites. RTM is a useful asset for guaranteeing that experiments cover the entirety of the prerequisites.

To validate the test object:

Testing ensures the execution of necessities just as the confirmation that they work as expected by clients. Approval is the term for this sort of testing. It is the way toward assessing an item after it has been created. Approval may be done physically or consequently. It as often as possible uses an assortment of testing draws near, like Black Box, White Box, etc. Approval is frequently performed by analyzers, despite the fact that clients can likewise approve the item as a component of User adequacy testing. The buyer is viewed as the lord in any firm.

To build confidence:

Improved programming quality is quite possibly the main objectives of programming testing. There are less defects in excellent programming. To put it another way, the more compelling the testing interaction is, the less blames will be found in the last yield. Thus, the general nature of the test thing will improve. Magnificent quality prompts higher consumer loyalty and less upkeep costs.

To prevent defects:

One of the objectives of programming testing is to keep away from issues right off the bat in the advancement interaction. The cost and exertion of identifying issues early sets aside a great deal of cash and time. Imperfection anticipation involves doing a main driver of recently found issues and afterward finding a way explicit ways to stay away from comparative blunders from happening later on. Powerful testing helps in the making of a blunder free item. At the point when flaws are forestalled, the general imperfection include in the item is decreased, bringing about a greater item for the purchaser.

To find defects in product:

Another significant objective of programming testing is to discover all imperfections in an item. The objective of testing is to uncover however many imperfections as could be expected under the circumstances in a product item while likewise guaranteeing that the application addresses the client's issues. Deformities ought to be found as right on time as practical in the testing cycle.

To share information to stakeholders:

The objective of testing is to give partners the entirety of the data they need concerning specialized or different imperatives, hazard factors, confounding prerequisites, etc. It may appear as testing exercises or testing results that feature what's missing and what wasn't right.

To reduce the level of risk:

1.3 Hazard is a word which intends to the opportunity from losing. Objective of this testing is for bringing down the opportunity of a danger happening. On the off chance that we don't deal with these vulnerabilities, potential dangers will emerge during the advancement time frame, yet additionally all through the item's entire life cycle.

1.4 Benefits of Software Testing

- Cost-Effectiveness: This is the main advantages of programming testing. Testing on schedule of any task permits us to save cost later on. It is more affordable to address issues found from the get-go in the product testing measure.
- Security: Software testing's generally helpless and delicate benefit is security. Individuals are looking for things that they can trust. It helps in the early identification of perils and issues.
- Product quality: Any product item should meet this basis. Testing ensures that purchasers get a top notch item.
- Customer Satisfaction: The essential objective of each item is to give consumer loyalty. The ideal client experience is guaranteed through UI/UX testing.

1.5 Test Process in Software Testing

1) Testing is all the more a system than a solitary undertaking. Testing should be arranged, and finishing it requires discipline. The nature of the test technique utilized decides the quality and viability of programming testing. The testing interaction might be separated into the accompanying essential advances:

2) Analysis and Design

- The following are the critical errands of test examination and test plan:
- To go through the testing premise once more. The test base is alluded to as information concerning experiments like necessities, exhaustive plan, item hazard investigation, engineering and interfaces.

- Test conditions to be resolved
- Tests are made
- Planning the test climate and recognizing the innovations and devices vital.

3) Implementation and Execution

4) Evaluating Exit criteria and Reporting

Leave standards are utilized to decide when testing ought to be halted. It is dictated by code inclusion, usefulness, and hazard. All in all, it is influenced by market dangers, cost, and idealness, and differs from one undertaking to another. At the point when the accompanying conditions are met, leave measures are utilized:

- A specific level of experiments are finished with a particular pass rate.
- The bug rate falls under a specific limit
- At the point when we comply with our time constraints

5) Test Closure activities:

At the point when the product is fit to be conveyed, test conclusion tasks are finished. Testing can be ended for an assortment of reasons, including:

- the crossing out of an undertaking
- the accomplishment of a specific objective
- When there is an upkeep discharge or an update

Coming up next are the critical obligations of test conclusion exercises:

- Check that all planned expectations have been given and that every single usefulness in reports have been tended to.

- Giving over the test gear to the support group. They will help with the product.
- To survey the testing interaction and draw exercises for future deliveries and undertakings.

1.6 Levels of Testing

1.Unit testing: decides if programming segments proceed true to form.

2.Integration Testing: analyzes the progression of information starting with one module then onto the next.

3.System Testing: evaluates the two kinds that is utilitarian and non-useful testing .

4.Acceptance Testing: Verifies that a determination or agreement's models are fulfilled as per its conveyance.

Every one of these layers of testing has an unmistakable reason. The product advancement lifecycle profits by these degrees of testing.

1) Unit testing: Unit is the individual segment of a framework or program that can be assembled, stacked, and run. This kind of testing permits every module to be tried autonomously.

The objective is to test every segment of the product independently. It decides if the parts are playing out their capacities. Engineers are the ones who do this kind of testing.

2)Integration testing: The expression "mix" alludes to the way toward blending at least two , like separate programming modules are joined and after that they are tried in a gathering in this testing step to guarantee that the associated framework is prepared for testing stage. Analyzers are the ones that do this sort of testing.

3)System testing: This is done on a practical, incorporated framework. It empowers for the confirmation of the framework's consistence with the principles. It inspects the general interaction of the different segments. Burden, proficiency, soundness, and security testing are all important for the interaction. Framework testing is regularly the last advance during the time spent guaranteeing that the framework fits the necessities. It surveys both useful and non-practical testing necessities.

4)Acceptance testing: This sort of testing utilizes legitimate strides for deciding whether the standards of a determination or agreement have been fulfilled as of the date of conveyance. The client or client is

the person who does acknowledgment testing. Different investors, then again, might be dynamic all the while.

Kinds of acknowledgment testing:

Alpha testing:

This is another kind of testing that searches for programming's imperfections and blemishes. This sort of test is done towards the finish of the application advancement, before the item is dispatched or conveyed to the client, to check that the client or customer gets a mistake free programming program. Alpha testing precedes beta testing, consequently you'll have to run beta testing whenever you've finished alpha testing.

Alpha testing doesn't occur in a true setting. Maybe, these evaluations are completed in a virtual climate that intently looks like the real world.

Beta testing :

Beta testing follows alpha testing, as recently expressed. Before an item is delivered, it is exposed to beta testing. It is done in a genuine setting by few enlisted clients and clients to guarantee that the program is sans blunder and runs appropriately. A few enhancements are performed to improve the program subsequent to gathering input and productive analysis from those clients. Subsequently, while programming is in beta testing, it is alluded to as a beta form. After this round of testing is finished, the program is made accessible to the overall population.

1.7 Types of testing

1.7.1. FUNCTIONAL TESTING

It is a kind of programming testing where the product framework is approved against the useful determination or necessities. Practical tests are utilized to approve the yield of a product program by giving sufficient information and contrasting it with the useful prerequisites.

Objective of this testing

- Fundamental Usability: It involves essential convenience testing of the framework.

- Mainline Functionality: It involves testing the essential elements of an application. It assists the analyzer with finding if a client could peruse across the screens absolutely or with no issue.
- Accessibility: It affirms that the framework is effectively open to the client.
- Error Conditions: Error conditions are checked utilizing testing strategies. It inspects if the fitting blunder messages are introduced.

1.6.2 NON-FUNCTIONAL TESTING

It is a sort of programming test that explores the non-useful components of a product program (execution, ease of use, constancy, etc).

Destinations of this testing

- Nonfunctional testing should assist with improving the program convenience, effectiveness, practicality, and conveyability.
- Assists in bringing down a danger or cost of non-useful pieces of the item all through creation.
- Improve the technique wherein the item is downloaded, designed, run and kept up
- Gain a superior comprehension of how items act and what innovations are being used.

Qualities of Non-practical testing

- This testing should be quantitative, hence emotional qualities like great, better, best, etc should not be utilized.
- Exact figures are probably not going to be found at the start of the prerequisite interaction.
- Prioritizing every single need is basic.
- Ascertain that quality ascribes are precisely perceived in Software Engineering.

1.6.3 Black Box Testing

BlackBox Testing is an assessing programming strategy which incorporates testing programming applications functionalities without understanding the code structure, execution subtleties or inner pathways. Discovery Testing is a kind of programming testing which centers around programming items yields and inputs and is completely determined by programming details. Another name for it is social trying.

Figure1.3 Black Box testing

Black Box Testing Techniques

- **Proportionality Class Partition:** This method is planned to keep the measure of plausible experiments to a base while yet guaranteeing satisfactory inclusion.
- **Limit Value Analysis:** BVA likewise checks the qualities at the edges. This method surveys if the scope of qualities is satisfactory to the framework. It's an extraordinary method to eliminate the measure of experiments. It's best for frameworks with inputs that fall between explicit reaches.
- **Choice Table:** A choice table is a network that associates causes and results. Every section has an alternate blend.
-

1.6.4 White Box Testing

This is another kind of testing which is a product survey technique that contains testing the fundamental construction, engineering and code of the item to assess the progression of info and improve the plan, convenience and security. Clear box testing, open box testing, straightforward box testing, code-based testing, and glass box testing is now and again known as white box testing since the code is noticeable to analyzers.

Due to the transparent box idea, the epithet "White Box" was authored. The expression "clear box" or "White Box" alludes to the capacity to investigate the product's internal operations by means of its outside shell (or "box"). Essentially, the "black box" in "Discovery Testing" signifies the powerlessness to notice the product's internal functions, permitting just the end-client experience to be surveyed.

Confirmation of this testing

• Vulnerabilities in inside security

• Breaking or severely planned coding ways

•The stream of specific sources of info through the code

•Expected yield

•Conditional circle usefulness

• Each assertion, article, and capacity should be tried independently.

White Box Testing Techniques

Code Coverage Analysis is a well known White box testing approach. A Test Case suite's openings are filled through Code Coverage investigation. It recognizes portions of a program that aren't scrutinized in an assortment of experiments.

Code inclusion investigation might be performed utilizing robotized advances. A crate analyzer can utilize the accompanying inclusion investigation strategies:

Articulation Coverage: During the testing of code , this system commands that each possible assertion present in the code should be tried in any event once.

Branch Coverage - This technique analyzes each possible way of a product application (if-else and other restrictive circles).

Benefits of White Box Testing

•Optimizing the code by identifying covered up flaws.

•Test situations for white boxes are not difficult to robotize.

•Testing is generally more broad since each code way is covered.

•Tests may begin from the get-go in the SDLC regardless of whether a GUI isn't accessible.

Weaknesses of White Box Testing

•Testing the white box may require some serious energy and cash.

•Developers used to test situations of white box hate it. The absence of data from engineers in the testing of white cases may prompt challenges underway.

•Professional assets are needed for white box testing with a strong comprehension of programming and execution.

•Testing white box requires some investment; greater application programming needs more opportunity for complete testing.

1.6.5 Dynamic Testing

Dynamic testing is a kind of programming testing that inspects the unique conduct of programming. The fundamental objective of dynamic testing is to uncover imperfections in the program runtime climate by assessing programming conduct with dynamic factors or factors that are not consistent. To test the powerful conduct, the code should be run.

Dynamic testing Goals

The essential objective of this sort of testing is to check if the product capacities precisely both when introducing it, bringing about a dependable program liberated from extreme issues. The significant objective of the unique test is to ensure that the program is reliable. Consistency identifies with an assortment of prerequisites like execution, convenience, combativity, etc, making Dynamic Testing incredibly essential.

Dynamic Testing Advantages

•This sort of testing could unveil unseen blemishes that are excessively intense or complex to be caught by static testing.

•In this sort of testing, we run the program from starting till the end, guaranteeing code without blunder and hence improving the presentation of the product.

•This sort of testing has gotten basic for spotting security dangers.

Disadvantages of Dynamic Testing

•This sort of testing takes tremendous measure of time since it runs the application/programming or code, that burns-through a ton of resources.

•Because this sort of testing don't start right on time in the lifecycle of programming, any mistakes that are rectified later in the process may raise the expense of the venture/item.

1.6.6 Static testing

This kind of testing is a product testing procedure that searches for mistakes without running the code in the program. It is mostly utilized in getting shortcomings in the beginning of advancement measure, when they are simpler to spot and address. It likewise helps the analyzer in the discovery of deficiencies that Dynamic Testing will be unable to distinguish. Dynamic Testing, then again, analyzes an application while the code is running. These are essential methods of static testing:

•Testing program physically: Tests done physically, regularly called as surveys, incorporate manual code examination.

•Tool-helped robotized investigation: Tool-helped computerized examination is basically static investigation.

Execution of Static Testing

1.Validation of Use Cases Requirements: This guarantees that all end-client exercises, just as any information/yield, are recognized. The experiments can be more precise and complete if the utilization cases are more express and complete.

2.Validation of Functional Requirements: This guarantees that all significant perspectives are recognized in the Functional Requirements.

3.Architecture Review: This incorporates all business-level cycles like worker areas, network graphs, convention particulars, load adjusting, data set openness, test hardware, etc.

4.Validation of Prototypes/Screen Mockups: This progression involves the check of and use cases.

1.6.7 Confirmation testing

Affirmation testing is a kind of programming testing where analyzers retest a product item to guarantee that recently revealed absconds have been fixed or are not, at this point present. At the point when a test fizzles, analyzers for the most part report a deformity. Testing group will then retest to check whether the detailed issue has been tended to. Affirmation testing is the term for this. Re-testing is another term for affirmation testing. To copy the issue, analyzers should allude to the imperfection report that was arranged when the issue was first detailed. The imperfection report helps the analyzer in playing out the test by guaranteeing that a similar test measures, test information, and test climate are utilized.

Execution of Confirmation Testing

Affirmation testing is completed under the accompanying circumstances inside the product testing life cycle.

1. Whenever the advancement group delivers another form with bug fixes

2. Verify the bug fix prior to doing relapse testing.

3. When the improvement group dismisses an issue report from an analyzer. To repeat the issue, analyzers do affirmation testing.

1.6.8 Regression testing

It is a kind of programming testing utilized for guaranteeing that a program or code alteration hasn't broken current functionalities. Relapse Testing is only a total or incomplete re-execution of recently performed experiments to affirm that current usefulness is working appropriately. This testing guarantees that new code adjustments don't have potentially negative results for current usefulness. It ensures that the old code keeps on working after the latest code adjustments have been made.

Need of Regression Testing

This sort of testing arises regularly when there is a need to modify the given program and furthermore on the off chance that we need to check whether the changed program impacts various parts in the given

program. It is additionally required when an engineer adds or changes highlight of a program, just as for bug and execution issues.

Relapse Testing Tools

Relapse testing costs will rise if your product is regularly refreshed. Testing the application physically builds the hour of execution of test and costs in such conditions. In such

examples, robotizing the product utilizing this apparatuses is the most ideal choice. The measure of robotization is controlled by the quantity of experiments that might be reused in ensuing relapse cycles.

1.Selenium: It is the opensource device for web application robotization. Selenium is a program based relapse testing apparatus.

2.Quick Test Professional (QTP): QTP is a robotized programming that might be utilized to computerize practical and relapse test situations. This computerizes utilizing VBScript programming. It's a watchword driven, information driven instrument.

3.Rational Functional Tester (RFT): This is a Java device for robotizing programming application test situations. It interfaces with Rational Test Manager and is generally utilized for computerizing relapse test situations.

Difficulties in Regression Testing:

The most widely recognized relapse issues:

•Test suites fill in size as more relapse runs are performed. The entire relapse test suite can't be arrived behind schedule to time and monetary limitations.

•Reducing the test situations while keeping up most extreme measure of inclusion is as yet the issue.

•Determining the recurrence of these Tests, like later every modification, assemble update, or set of issue fixes, is difficult.[4]

1.6.9 Smoke Testing

This is a product testing strategy which surveys whether a framework bundle is conveyed and is looking after strength. "Construct Verification Testing" or "Certainty Testing" are different terms for smoke testing. To put it another way, we're verifying whether the primary highlights are working and if there are any masterpieces in the form we're trying.

This permits you to check whether the form is flawed, keeping you from sitting around and cash on additional testing. The smoke tests show that the construction is prepared for additional conventional testing. The essential objective of smoke testing is to uncover extreme issues from the beginning. Smoke tests are utilized to show that a framework is steady and meets the models. All information records, libraries, reusable modules, and designing parts important to achieve at least one item functionalities are remembered for a form.

Execution of this testing

The advancement group conveys the form in QA in this testing method. Analyzers perform experiments on the form subsequent to stepping through determinations of examination cases. The application's fundamental usefulness is tried by the QA group. The reason for this arrangement of experiments is to uncover development mistakes. On the off chance that these tests pass, the QA group will proceed onward to Functional Testing.

Need of smoke testing

Smoke testing is indispensable in programming advancement since it guarantees the framework's precision in the beginning phases. We can set aside time and cash thusly. As a result, smoke tests guarantee that the framework is in legitimate working request. Just when we've finished smoke testing will we start practical testing.

- Smoke testing will recognize the entirety of the form's masterpieces, and it will be done after the form has been delivered to QA. Smoke testing distinguishes most of blunders during the beginning stages of programming advancement.

•Smoke testing makes recognizing and adjusting serious imperfections a lot simpler.

- Smoke testing permits the QA group to distinguish application usefulness blemishes that may have created because of the new code.

- Smoke testing recognizes the most genuine mistakes.

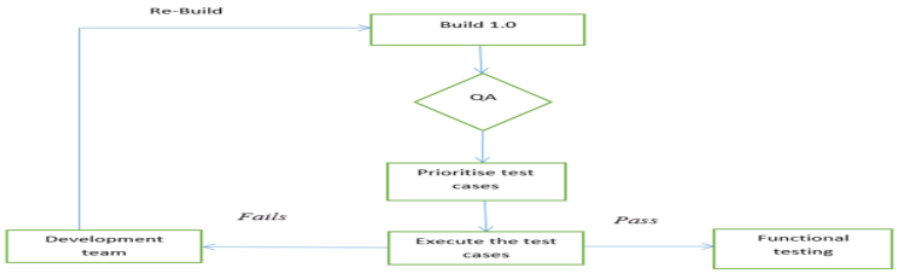


Figure1.4 Smoke test cycle

1.8 Test case writing

It is an arrangement of steps performed for guaranteeing that the specific usefulness or ability of our product program is working appropriately. Experiment implies an assortment of test strategies, information, preconditions, and postconditions made for a specific test situation all together for approving any necessity. The experiment contains determined conditions that an analyzer may use to count between expected outcomes and genuine outcomes all together for evaluating if the application addresses the client's issues.

1.7.1 Test Case Management Tools

These are mechanization arrangements which aid the organization and upkeep of Test Cases.

1. Documentation of Test Cases: You may use instruments to accelerate the development of Test Cases by utilizing formats.

2. Run the experiment and monitor the outcomes: The instruments might be utilized for running a test , and the yields might be helpfully recorded.

3. Automate Defect Tracking: The tests which don't pass are quickly associated with a bug tracker, thereafter it tends to be allotted to coding group and finished up on email alarms.

4. Traceability: Requirements, Test cases, and Test case execution are completely associated by the devices, additionally every experiment can measure up to the past one to assess the number of experiments are covered.

.

1.7.2 Benefits of Writing Test Cases

The fundamental objective of an experiment is to ensure that different functionalities of an application proceed as arranged. It helps analyzers in deciding whether the program is sans deformity and meets the prerequisites of end clients. Different benefits of experiments include:

- guaranteeing great test inclusion,
- assisting with upgrading programming quality, etc. Diminishes the expense of upkeep and programming support
- Help with guaranteeing that the program fulfills the necessities of the end client.
- Permits the analyzer to contemplate the tests and approach them from however many points of view as could reasonably be expected.
- Experiments are reusable later on since they might be referred to and executed by anyone.

1.7.3 Test Case Format

An ID, depiction, a lot of data, two or three vital exercises, and expected and genuine outcomes are the rule portions of an analysis.

- Experiment Name: An investigation should have a quite obvious name or title.

- Experiment Description: two or three words, the depiction should portray the analyzer what they'll test.

- Pre-Conditions: Include any speculations that apply to the test similarly as any preconditions that ought to be satisfied before the test can be run.

- Experiment Steps: The test steps should fuse the aggregate of the crucial data similarly as rules for running the test. The stages should be clear and compact, without notwithstanding huge information. Test Data: It's essential to pick a data arrangement that gives adequate incorporation. Pick a data combination that fuses extraordinary just as deplorable possible results.

- Expected Result: The ordinary outcomes depict what the analyzer should hope to see on account of the test strategies.

- Genuine Result: They portray how the application performed during the execution of investigations. Comments: Any additional information, for instance, screen catches, that the analyzer wishes to share might be added here.

This is the standard game plan used by analyzers when forming tests. Additional limits, for instance, test need, kind of analysis, and bug earnestness, can be added to these limits by analyzers.

Table 1.1: Test Case Example

Title	**Login with a valid username & password**
Precondition	User is already registered using valid credentials
Test Steps	1. Enter a valid username 2. Enter a valid password 3. Click on sign in
Expected Result	-User is logged in successfully -User is redirected to Home page Test Suite Login Test
Environment	Samsung Galaxy Note 10 - Android 10 – 4G Network

Actual Result	Same as expected
Status	Pass

1.9 Bug Report in Software Testing

In programming testing, a bug report is an exhaustive archive portraying the blemishes distinguished in the product program. A bug report includes everything about a bug, like the depiction, the date the bug was found, the character of the analyzer who found it, the name of the engineer who remedied it, etc. A bug report helps in the location of comparable issues later on, permitting them to be stayed away from.

The accompanying data ought to be remembered for the Problem Report when detailing the bug to the designer.

- Defect ID - The imperfection's special recognizing number.

- Defect Description-A nitty gritty portrayal of the Defect, remembering subtleties for the module where the Defect was found.

- Version - The program adaptation in which the defect was found.

- Steps - a point by point set of steps with screen captures that the engineer may use to imitate the defects.

Date Raised - Date when the deformity is raised

- Reference-where you give references to papers like as details, plan, design, or even pictures of the issue to help understand the shortcoming

- Detected By - The name/ID of the analyzer who revealed the blemish.

Status - Status of the deformity

Table 1.2: Defect Status

New (Ready to test)	The test case is not executed
Pass	The test case is executed and the actual result is the same as the expected result
Fail	The test case is executed and the actual result is different from the expected result
Blocked/Skipped	The test case can't be executed

- Fixed by - the engineer who settled it

- Date Closed - the date when the shortcoming was fixed

- Severity - the seriousness of the shortcoming's effect on the application

- Priority, which is attached to the earnestness of settling surrenders. As per the impact criticalness with which the issue ought to be redressed, the seriousness need may be High, Medium, or Low.

Table 1.3: Defect Report Example

Title	**Login -> Forgot password button isn't working**
Steps to reproduce	1. Click on Login 2. Click on Forgot password
Expected Result	The button can be clicked and user should be redirected to a page to enter his email
Actual result	Clicking on the button doesn't have any impact
Test Environment	Samsung Galaxy Note 10 - Android 10 – 4G Network
Priority	High
Type	Functional
Severity	High

1.8.1 Bug life cycle

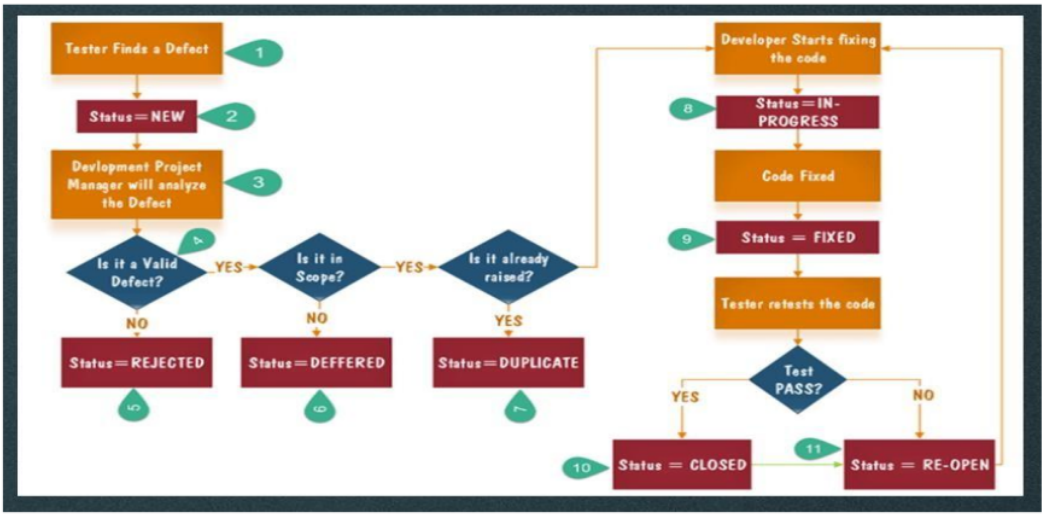


Figure1.5 Bug Life Cycle

CHAPTER 2

DATASOURCE

2.1 SQL

SQL is a relational database programming language for obtaining and managing data. The abbreviation for Structured Query Language (SQL) stands for Structured Query Language.

Structured Query Language (SQL) is a computer language for storing, manipulating, and retrieving data from a relational database.

SQL is the standard language for Relational Database Systems (RDBMS). All Relational Database Management Systems (RDMS) Server employ SQL as their main database language.

2.1.1 Applications of SQL

As recently said, SQL is perhaps the most broadly utilized information base question dialects.

- Provides admittance to information put away in social data set administration frameworks to clients.

- Allows clients to clarify information base.

- SQL modules, libraries, and pre-compilers might be utilized to insert in different dialects, permitting clients to characterize and adjust information in a data set. Clients may make and eliminate data sets and tables with this program.

- Provides clients with the capacity to make table inquiries, data set items, and activities.

- Users can change the limitations for sections, capacities, and articles.

2.1.2 SQL Process

In the event that we play out a Sql question towards any RDBMS, our product chooses the ideal way to deal with satisfy given solicitation and the SQL motor decides when to fathom it. There are a few parts to this activity. These are the components:

- Engines for enhancement

- Dispatcher of Query

- Search Engine (Classic)

- SQL Query Engine, for instance.

A customary question motor handles non-SQL inquiries, though a SQL inquiry motor handles consistent documents. The SQL Architecture is portrayed in the figure underneath.

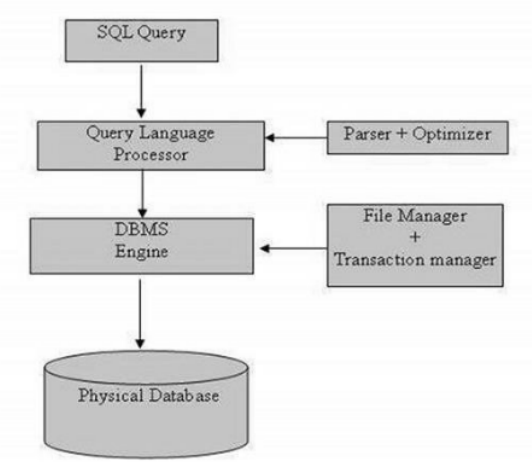


Figure2.1 SQL Architecture

2.1.2 RDBMS

"Social Database Management System" is an abbreviation for "Rdbms." A social data set administration framework, or RDBMS, is a data set administration framework made only for social data sets. RDBMS are along these lines a subset of DBMS.

It is one that utilizations lines and segments to store information in a coordinated manner. This makes finding and getting to specific qualities inside the data set a breeze. As its qualities inside each field are associated to each other, it is classified "social." Tables can likewise be connected to each other. In view of the social nature, inquiries might be directed on a few tables simultaneously.

2.2 XML

The shortened form for (XML) means "Extensible Markup Language." Despite Html components, which indicate how the information ought to be appeared, XML labels investigate the data and can be utilized for putting away and keeping up it as opposed to determining how it ought to be appeared. Sooner rather than later, XML would not wipe out HTML, however this will get to more data by joining a large number of HTML's astounding qualities.

Three fundamental qualities of XML make it valuable in a wide scope of frameworks and

arrangements:

- XML is extensible: We may plan our labels giving our own depiction and in any language, to accommodate our product.

XML moves information yet won't show it. For what reason does XML convey information yet not show it? You can save information in XML independent of how it very well may be shown.

- XML is a uninhibitedly available norm. The World Wide Web Consortium (W3C) created XML, that is free and open innovation.

2.2.1 Usage

XML could work behind at scenes to facilitate the age of HTML content for colossal sites, as indicated by a concise rundown of its employments.

Data might be sent among associations and organizations utilizing this. XML might be utilized to dump and reload data sets.

Information might be put away and coordinated utilizing XML, makes it simple to tweak specific social data set requirements.

XML in addition to templates might be effectively coupled to give practically any right result. Practically any sort of information might be sent utilizing a XML archive.

2.2.2 Markup Description

XML is a markup language for portraying of rules for order perceptions in a manner that is both physical and computerized comprehensible. Markup is data that is attached to something like an archive to improve its significance surely, for example, recognizing the components and their connections. It is a bunch of images that might be placed into a record's book to recognize and mark its various bits.

2.3 JSON

JSON, or JavaScript Object Notation, is a book based open norm for moving comprehensible information. Software engineers know about JSON shows and other programming dialects.

- JSON design was made by Douglas Crockford

- It was made to trade information in a comprehensible arrangement.

- The JavaScript programming language has been extended.

- The .json record expansion is utilized.

- JSON is a kind of information design. Application/json is the Internet Media Type.

- The Uniform Type Identifier (UTI) is available to general society. json is a sort of information.

2.3.1 Uses of JSON

- It's used in the improvement of uses which are java based, like augmentations of program and website pages.

- It is a serialization or transmission design for information which is organized over an association of organizations.

- This is for the most part utilized for send data from a worker to site applications.

- JSON is the arrangement utilized by online administrations and APIs to uncover information which isn't private.

- This is viable by the present programming dialects.

2.2.1 JSON Characteristics

- It is basic in comprehension and composing.

- This is a tradable arrangement which depends on text and that is not difficult to utilize.

- JSON is language rationalist.

CHAPTER 3

VBSCRIPT

Visual Basic Scripting, or VBScript, is a subset of Visual Basic for Applications (VBA). This is a Microsoft item that might be found not simply in Microsoft items like MS Project and MS Office, and can be utilized in various projects like AUTO CAD.

3.1 VBScript highlights

- It's a basic prearranging language with a lightning-quick processor.

- VBScript is case-unfeeling generally. It has an exceptionally straightforward grammar that is not difficult to get and use.

•VBScript isn't an Object-Oriented Programming language like C++ or Java, yet rather an item based prearranging language.

- To access the parts of the climate wherein it is running, it utilizes the Component Object Model (COM).

- Only in a Host Environment, like Internet Explorer (IE), Internet Information Services (IIS), or Windows Scripting Host, can VBScript be effectively run (WSH)

3.2 VBscript – Version History and Uses

Microsoft delivered the principal form of VBScript in 1996, and it was variant one. Latest stable form of VBScript is 5.8, which is incorporated with Internet Explorer 8 and Windows 7. Where VBScript might be utilized are various and are not restricted to the ones recorded underneath.

- Quick Test Professional, truncated as QTP, is a famous robotization testing application that utilizations VBScript as a prearranging language.

- Windows Scripting Host, which is for the most part utilized for computerizing the Windows Desktop by Windows System Administrators.

- Active Server Pages is a prearranging climate which has worker that utilizes VBScript or Java Script to make dynamic pages.

- In Microsoft Internet Explorer, VBScript is used for customer side prearranging.

•VBScript is utilized to execute Microsoft Outlook Forms, though VBA is utilized for application-level programming.

Disservices

- Only Internet Explorer programs support VBscript. VBScript isn't upheld by different programs, like Chrome and Firefox. Therefore, JavaScript overshadows VBScript.

- VBScript's order line usefulness is restricted.

•Debugging is extreme because of the absence of a default improvement climate.

VBScript use

The most recent adaptation of VBScript is 5.8, likewise with the most recent presentation of the.NET structure, Microsoft has selected to keep on supporting it for web improvement inside ASP.NET. Subsequently, no new forms of the VBScript motor will be delivered, anyway the Microsoft supporting Engineering Team will deal with any imperfection fixes and security issues. The VBScript motor, then again, would be incorporated naturally in all Microsoft Windows and IIS establishments.

CHAPTER 4
UFT AUTOMATION

Bound together Functional Testing, or UFT, is a computerized utilitarian testing instrument that helps analyzers in executing robotized tests to discover any slip-ups, blames, and spaces inside application under test that are not steady with the normal outcomes. Mercury Interactive made it, which was ultimately purchased by HP and is presently Microfocus. QTP represents Quick Test Professional, while UFT represents Unified Functional Testing.

UFT, initially Quick Test Professional (QTP), is a piece of programming that computerizes utilitarian and relapse testing for programming applications and conditions.

It has a graphical UI and supports catchphrase and prearranging interfaces. It indicates a test technique and controls the articles and controls of the application under test utilizing the Visual Basic Scripting Edition (VBScript) prearranging language. From a solitary reassure, designers may test each of the three layers of a program's activities: the interface, the assistance layer, and the data set layer.

Speedy Test Professional was the underlying name for UFT, which was made by Mercury Interactive. Hewlett-Packard (HP) in the long run purchased Mercury Interactive in 2006. Up until 2016, when the HP Software Division was offered to Micro Focus, UFT 11.5 coordinated HP Quick Test Professional and HP Service Test into a solitary programming bundle that was available through the HP Software Division.

4.1 DESCRIPTION

This is a kind of programming that mechanizes the trial of an assortment of programming applications and settings. It utilizes a UI, like a local GUI or a web interface, to do utilitarian and relapse testing. It works by perceiving objects in an application's UI or a page and performs determined exercises; it might likewise record object information like property and id of controller. To portray the test technique and

alter the articles and controls of the application under test, This product utilizes the VBScript prearranging language. Clients may have to change the hidden to execute more intricate activities.[7]

Exemption dealing with

Miniature Focus UFT handles exemptions through recuperation situations, with the reason for proceeding to execute tests regardless of whether there is an unforeseen disappointment. Since UFT interfaces further into framework memory of the projects being tried, different occasions may make HPE Unified Functional Testing cut short and be unrepairable.

Information driven testing

Zero in on the little subtleties Data-driven testing is empowered at UFT. Data could be sent out to a dominate sheet, for instance, and reused elsewhere. This is completed utilizing a Microsoft Excel worksheet which might be moved toward utilizing UFT. The Global information sheet and the Action (neighborhood) information sheet are the two sorts of information tables utilized by UFT. The test stages can take information from these information tables and drive variable information into the application under test, guaranteeing that an expected result is accomplished.

Mechanizing custom and complex UI objects

Zeroing in on the little subtleties Customized UI things and other confounded things might be missed by UFT. These things can be named virtual items or knowledge objects by clients. Virtual articles are not really upheld by UFT for simple account or recording in low-level mode.

Extensibility

This might be tweaked with additional items having a wide scope of improvement conditions which aren't conveyed outside the application. Web, .NET, Java, and Delphi are completely upheld by UFT add-ins. HP Functional Testing programming incorporates HP Quick Test Professional and the HP Quick Test

Professional add-ins. UI Design

It gives two perspectives on a test script, just as choices to modify it: Keyword View and Expert View. These perspectives empower UFT to work as a test's Integrated Development Environment (IDE), and UFT contains a few customary IDE abilities, for example, breakpoints, which permit you to end a test at specific focuses.

Clients may use Keyword View to construct or check test methodology in a particular, plain way. Each line shows a stage that can be changed. Any of the accompanying segments can be remembered for the Keyword View: Item, Operation, Value, Assignment, Comment, and Documentation. UFT shows a relating line of content for each stage in the Keyword View dependent on the worth of line and section.

Whenever, clients can add, erase, or modify steps.

Clients may likewise examine credits for objects like designated spots, yield esteems, and activities in Keyword View, just as use restrictive and circle articulations and supplement breakpoints to help troubleshoot a test.

It permits individuals to view or change the source code of a test utilizing VBScript. Analyzers can modify all test ventures aside from the root Global activity, and changes are adjusted with the watchword see, which is intended for more experienced clients.

The prearranging language utilized by Micro Focus UFT is VBScript. Classes are upheld by VBScript, anyway polymorphism and legacy are not. VBScript loses the capacity to use some Visual Basic watchwords, just as an incorporated debugger, an occasion controller, and a structures proofreader, when contrasted with Visual Basic for Applications (VBA). Despite the fact that HP has incorporated a debugger, it is confined in contrast with testing instruments that incorporate a full-highlighted IDE, for example, those included with other programming dialects.

4.2 Drawbacks

Miniature Focus UFT is a Windows-based application. It depends on fairly antiquated Windows-just innovation like ActiveX and VBScript, neither of which is an article situated language. This product unfit

to test with all program variants and sorts. It doesn't, for instance, support Opera.

Albeit far off execution is as yet accessible with HPE Unified Functional Testing running on a different framework, the Test Execution motor is combined with the GUI Test Code improvement IDE, accordingly it is extremely unlikely to run the tests freely of UFT.

Due to the high permit charges, the innovation is every now and again just used by a little gathering of analyzers inside an organization. This backings a compartmentalized way to deal with QA/testing, where test is done independently from the business and advancement groups, as opposed to a cooperative methodology where analyzers work together intimately with the engineers and business group.

CHAPTER 5
DESCRIPTION OF WORK CARRIED OUT

5.1 Writing test cases for hotel booking website

Test Scenario

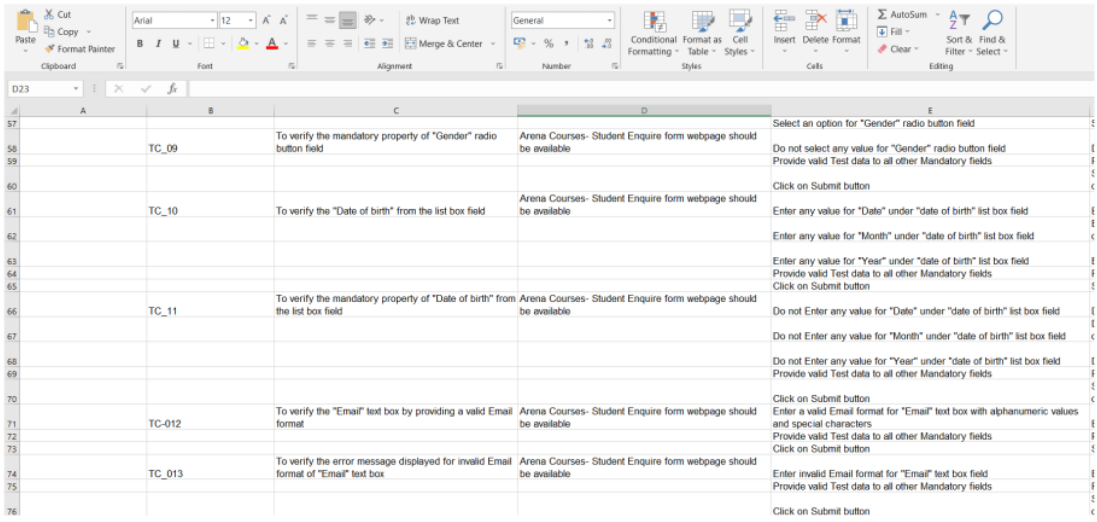
Module	Scenario ID	Scenario Name	Scenario Description	Requirement id
Arena Courses Website- Student Enquiry form	TS_01	Find Course- Valid	To check whether the user is able to find course if he/she provides valid details	RID_01
	TS_02	Find Course- Invalid	To check whether the user is able to find course if he/she provides invalid details	
	TS_03	Find Course- Null Values	To check whether the user is able to find course by not providing any details	RID_02
	TS_04	Find Course- Alternate Flow	To check whether the user is able to find course if he/she selects the cancel button	RID_03
	TS_05	Fields Validation	To Validate all the possible fields available for Submit functionality	

Test Cases

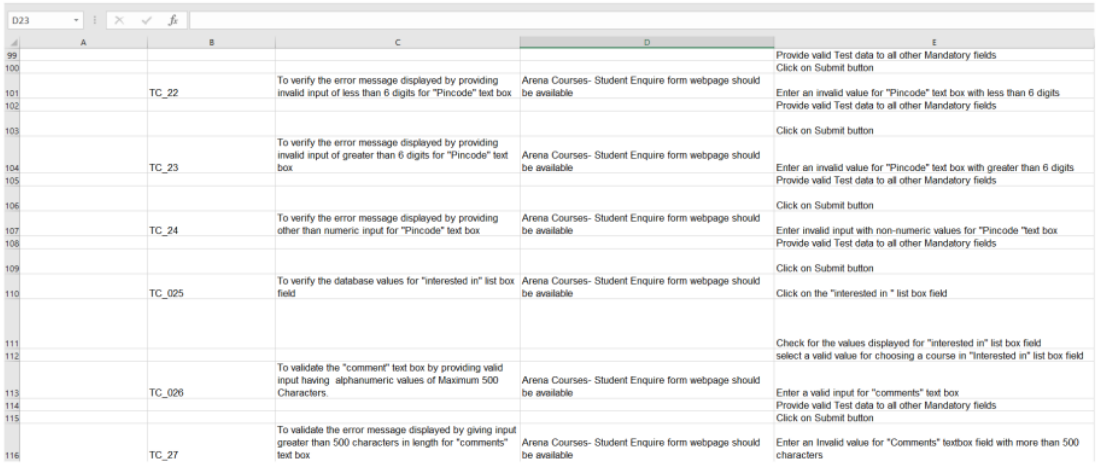
D23

	A	B	C	D	E
2	TS_01	TC_01	To verify the Submit Functionality by providing the valid inputs	Arena Courses- Student Enquire form webpage should be available	Step1. Enter a valid value for "Name" textbox field
3					Step2. Select "Gender" option for radio button field
4					Step3. Select the "Date of birth" from the list box in DD-Month-Year format
5					Step4. Enter a valid value for "Email ID" textbox field
6					Step5. Enter a valid value for "phone" textbox field
7					Step6. Enter a valid value for "city" textbox field
8					Step7. Enter a valid value for "Pincode" textbox field
9					Step8. Select a valid value for choosing a course in "Interested in" list box field
10					Step8. Enter a valid value for "Comments" textbox field
11					Step9. Click on the submit button
12					
13					
14	TS_01	TC_02	To check the **submit** functionality for the courses by filling the Invalid "Students enquiry form"	Arena Courses- Student Enquire form webpage should be available	**Enter an Invalid value for "Name" textbox field**
15					Select "Gender" option for radio button field
16					Select the "Date of birth" from the list box in DD-Month-Year format
17					Enter an Invalid value for "Email ID" textbox field
18					Enter an Invalid value for "phone" textbox field
19					Enter an Invalid value for "city" textbox field
20					Enter an Invalid value for "Pincode" textbox field
21					Select a valid value for choosing a course in "Inserted in" list box field
22					Enter an Invalid value for "Comments" textbox field

TS_02	TC_03	To Check the Submit functionality for the courses by filling null inputs in "Students Enquire form"	Arena Courses- Student Enquire form webpage should be available	Do not Enter a valid value for "Name" textbox field
				Do not Select "Gender" option for radio button field
				Do not Select the "Date of birth" from the list box in DD-Month-Year format
				Do not Enter a valid value for "Email ID" textbox field
				Do not Enter a valid value for "phone" textbox field
				Do not Enter a valid value for "city" textbox field
				Do not Enter a valid value for "Pincode" textbox field
				Do not Select a valid value for choosing a course in "Interested in" list box field
				Do not Enter a valid value for "Comments" textbox field
				Click on Submit button
TS_03	TC_04	To verify "Name" field by entering valid Name having only alphabets values of maximum 30 characters	Arena Courses- Student Enquire form webpage should be available	Enter a valid name into "Name" textbox field
				Provide valid Test data to all other Mandatory fields
				Click on Submit button
	TC_05	To verify "Name" field by entering non-alphabetic values	Arena Courses- Student Enquire form webpage should be available	Enter an Invalid name value for "Name" textbox field with non-alphabetic values
				Provide valid Test data to all other Mandatory fields

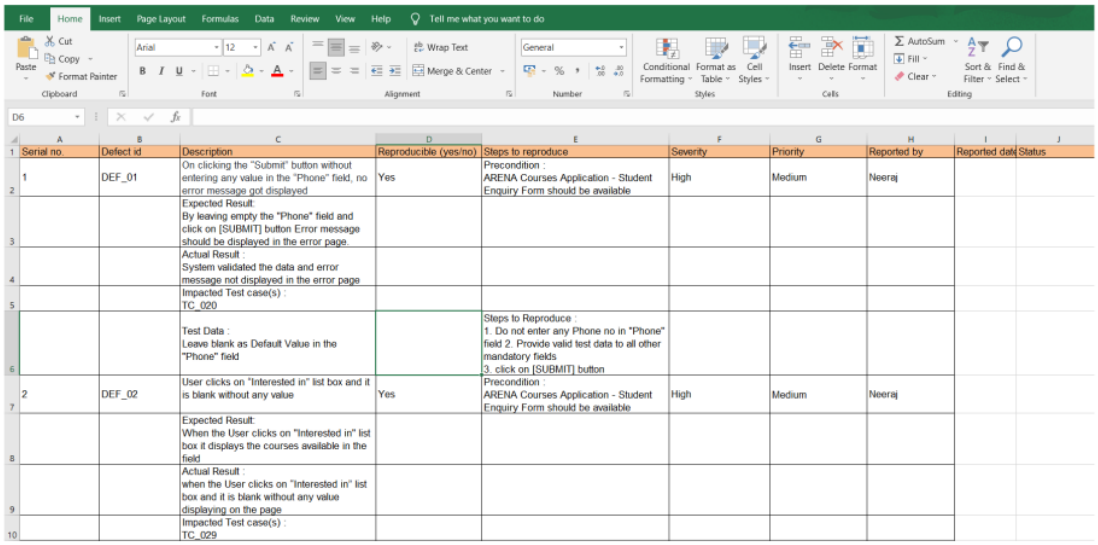


	A	B	C	D	E
57					Select an option for "Gender" radio button field
58		TC_09	To verify the mandatory property of "Gender" radio button field	Arena Courses- Student Enquire form webpage should be available	Do not select any value for "Gender" radio button field
59					Provide valid Test data to all other Mandatory fields
60					Click on Submit button
61		TC_10	To verify the "Date of birth" from the list box field	Arena Courses- Student Enquire form webpage should be available	Enter any value for "Date" under "date of birth" list box field
62					Enter any value for "Month" under "date of birth" list box field
63					Enter any value for "Year" under "date of birth" list box field
64					Provide valid Test data to all other Mandatory fields
65					Click on Submit button
66		TC_11	To verify the mandatory property of "Date of birth" from the list box field	Arena Courses- Student Enquire form webpage should be available	Do not Enter any value for "Date" under "date of birth" list box field
67					Do not Enter any value for "Month" under "date of birth" list box field
68					Do not Enter any value for "Year" under "date of birth" list box field
69					Provide valid Test data to all other Mandatory fields
70					Click on Submit button
71		TC-012	To verify the "Email" text box by providing a valid Email format	Arena Courses- Student Enquire form webpage should be available	Enter a valid Email format for "Email" text box with alphanumeric values and special characters
72					Provide valid Test data to all other Mandatory fields
73					Click on Submit button
74		TC_013	To verify the error message displayed for invalid Email format of "Email" text box	Arena Courses- Student Enquire form webpage should be available	Enter invalid Email format for "Email" text box field
75					Provide valid Test data to all other Mandatory fields
76					Click on Submit button

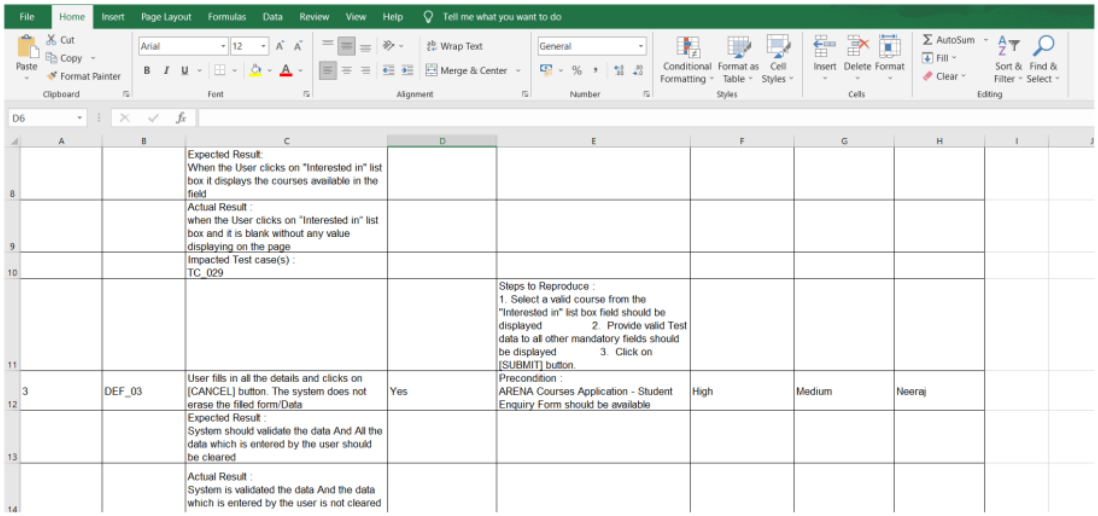


	A	B	C	D	E
99					Provide valid Test data to all other Mandatory fields
100					Click on Submit button
101		TC_22	To verify the error message displayed by providing invalid input of less than 6 digits for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter an invalid value for "Pincode" text box with less than 6 digits
102					Provide valid Test data to all other Mandatory fields
103					Click on Submit button
104		TC_23	To verify the error message displayed by providing invalid input of greater than 6 digits for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter an invalid value for "Pincode" text box with greater than 6 digits
105					Provide valid Test data to all other Mandatory fields
106					Click on Submit button
107		TC_24	To verify the error message displayed by providing other than numeric input for "Pincode" text box	Arena Courses- Student Enquire form webpage should be available	Enter invalid input with non-numeric values for "Pincode "text box
108					Provide valid Test data to all other Mandatory fields
109					Click on Submit button
110		TC_025	To verify the database values for "interested in" list box field	Arena Courses- Student Enquire form webpage should be available	Click on the "interested in " list box field
111					Check for the values displayed for "interested in" list box field
112					select a valid value for choosing a course in "Interested in" list box field
113		TC_026	To validate the "comment" text box by providing valid input having alphanumeric values of Maximum 500 Characters.	Arena Courses- Student Enquire form webpage should be available	Enter a valid input for "comments" text box
114					Provide valid Test data to all other Mandatory fields
115					Click on Submit button
116		TC_27	To validate the error message displayed by giving input greater than 500 characters in length for "comments" text box	Arena Courses- Student Enquire form webpage should be available	Enter an Invalid value for "Comments" textbox field with more than 500 characters

Defect Report

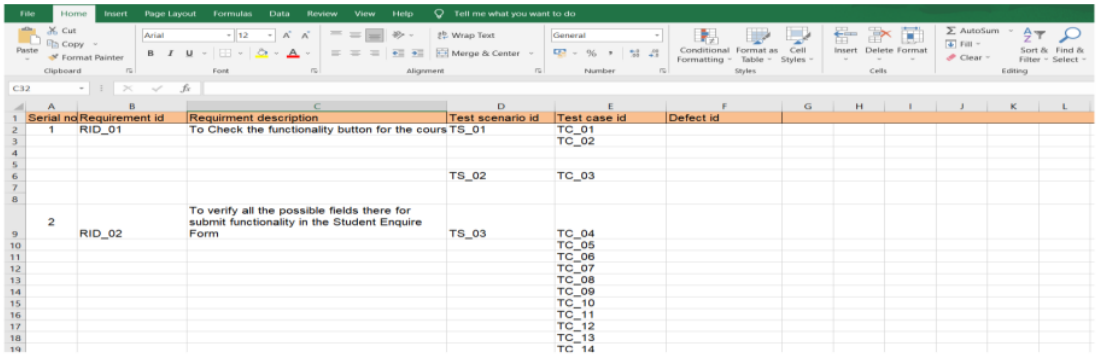


Serial no.	Defect id	Description	Reproducible (yes/no)	Steps to reproduce	Severity	Priority	Reported by	Reported date	Status
1	DEF_01	On clicking the "Submit" button without entering any value in the "Phone" field, no error message got displayed	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj		
		Expected Result: By leaving empty the "Phone" field and click on [SUBMIT] button Error message should be displayed in the error page.							
		Actual Result : System validated the data and error message not displayed in the error page							
		Impacted Test case(s) : TC_020							
		Test Data : Leave blank as Default Value in the "Phone" field		Steps to Reproduce : 1. Do not enter any Phone no in "Phone" field 2. Provide valid test data to all other mandatory fields 3. click on [SUBMIT] button					
2	DEF_02	User clicks on "Interested in" list box and it is blank without any value	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj		
		Expected Result: When the User clicks on "Interested in" list box it displays the courses available in the field							
		Actual Result : when the User clicks on "Interested in" list box and it is blank without any value displaying on the page							
		Impacted Test case(s) : TC_029							



		Expected Result: When the User clicks on "Interested in" list box it displays the courses available in the field					
		Actual Result : when the User clicks on "Interested in" list box and it is blank without any value displaying on the page					
		Impacted Test case(s) : TC_029					
				Steps to Reproduce : 1. Select a valid course from the "Interested in" list box field should be displayed 2. Provide valid Test data to all other mandatory fields should be displayed 3. Click on [SUBMIT] button.			
3	DEF_03	User fills in all the details and clicks on [CANCEL] button. The system does not erase the filled form/Data	Yes	Precondition : ARENA Courses Application - Student Enquiry Form should be available	High	Medium	Neeraj
		Expected Result : System should validate the data And All the data which is entered by the user should be cleared					
		Actual Result : System is validated the data And the data which is entered by the user is not cleared					

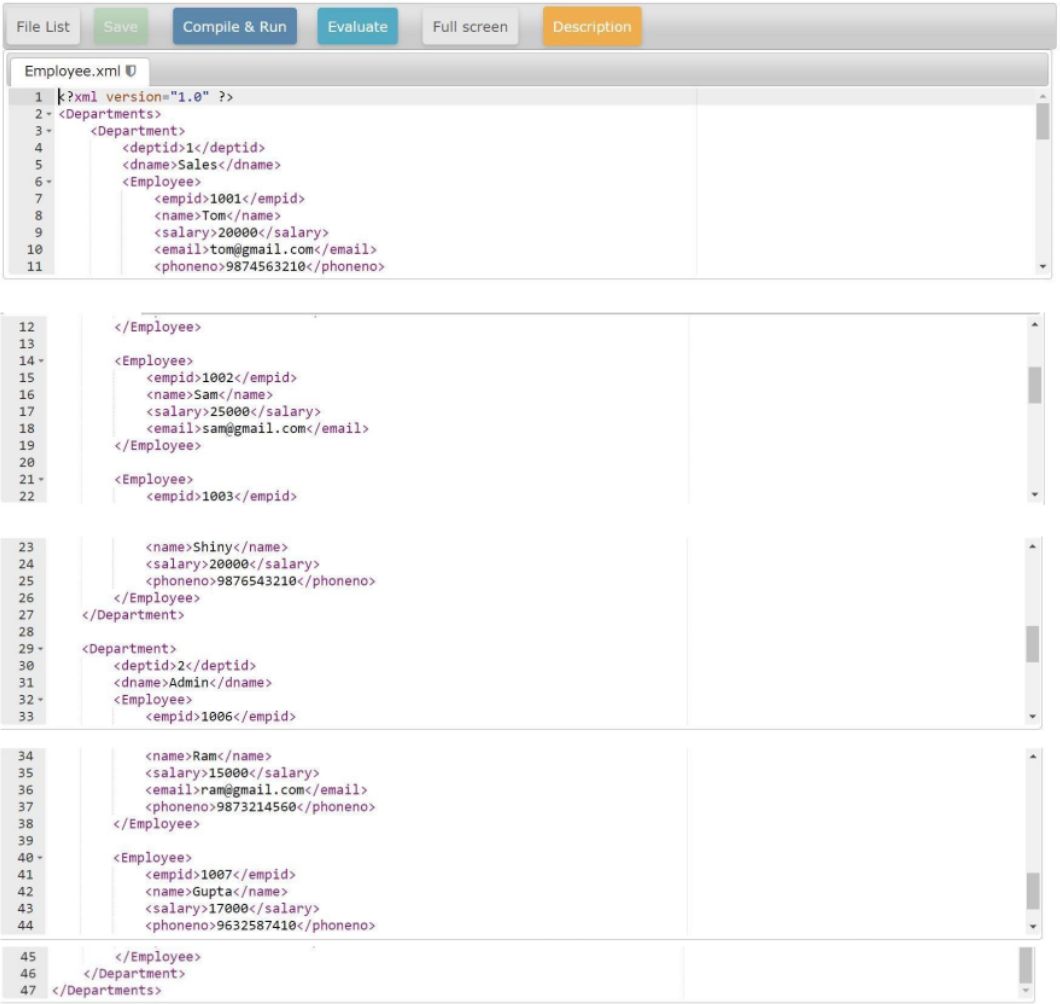
RTM (Requirement Traceability Matrix)



Serial no	Requirement id	Requirment description	Test scenario id	Test case id	Defect id
1	RID_01	To Check the functionality button for the cours	TS_01	TC_01	
				TC_02	
			TS_02	TC_03	
2	RID_02	To verify all the possible fields there for submit functionality in the Student Enquire Form	TS_03	TC_04	
				TC_05	
				TC_06	
				TC_07	
				TC_08	
				TC_09	
				TC_10	
				TC_11	
				TC_12	
				TC_13	
				TC_14	

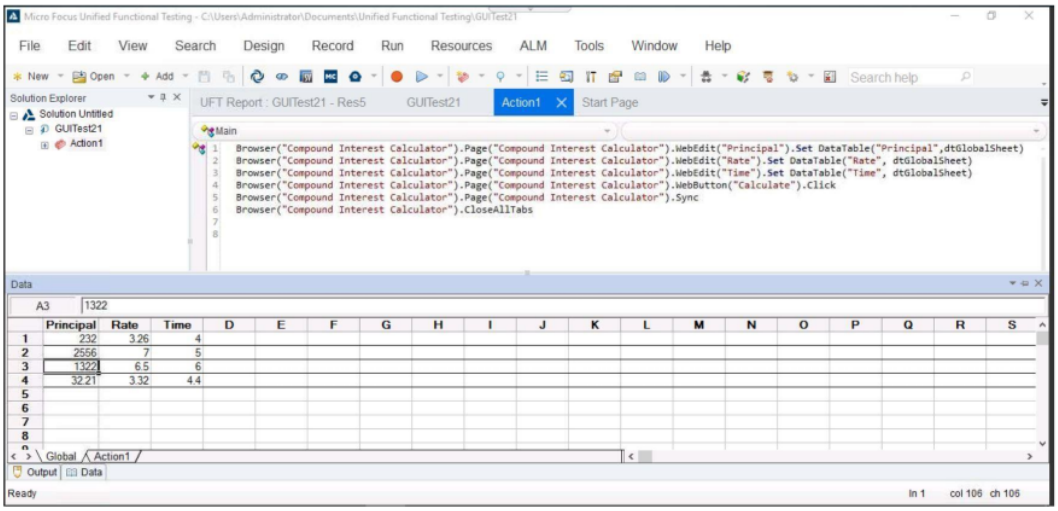
5.2 Creating a well-formed document for the given scenario using XML

Deptid	Dname	Empid	Name	salary	email	phoneno
1	Sales	1001	Tom	20000	tom@gmail.com	9874563210
1	Sales	1002	Sam	25000	sam@gmail.com	
1	Sales	1003	Shiny	20000		9876543210
2	Admin	1006	Ram	15000	ram@gmail.com	9873214560
2	Admin	1007	Gupta	17000		9632587410

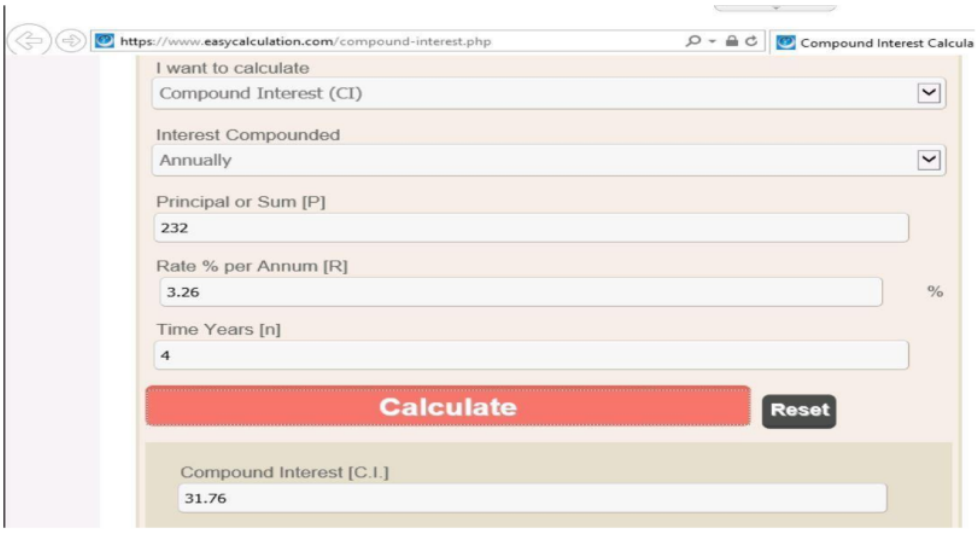


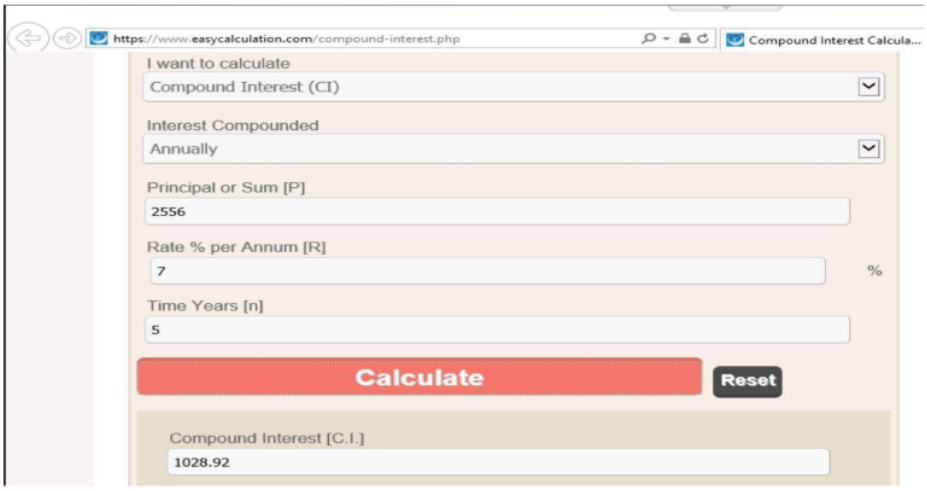
5.3 To check the functionality of calculate button for website http://easycalculation.com/compound-interest.php for different inputs using UFT Automation.

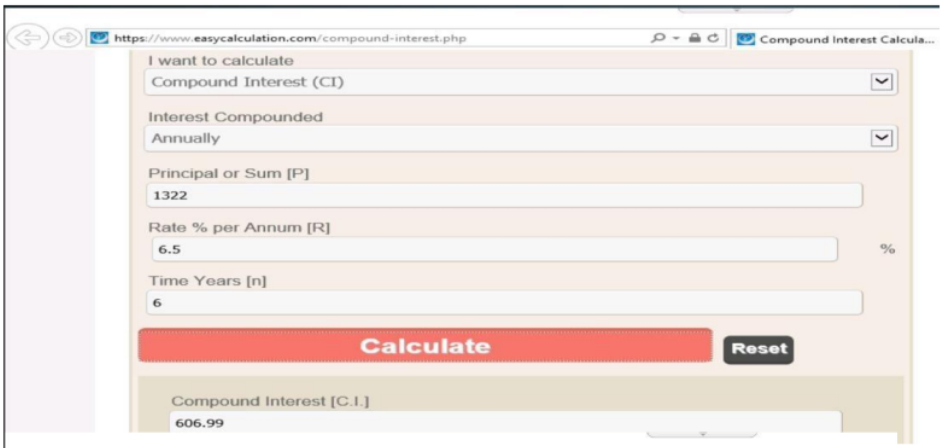
Script

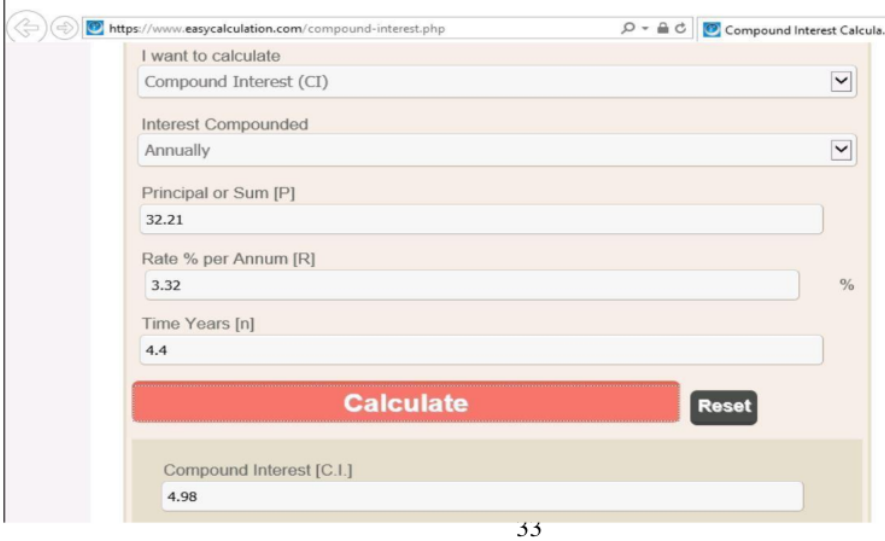


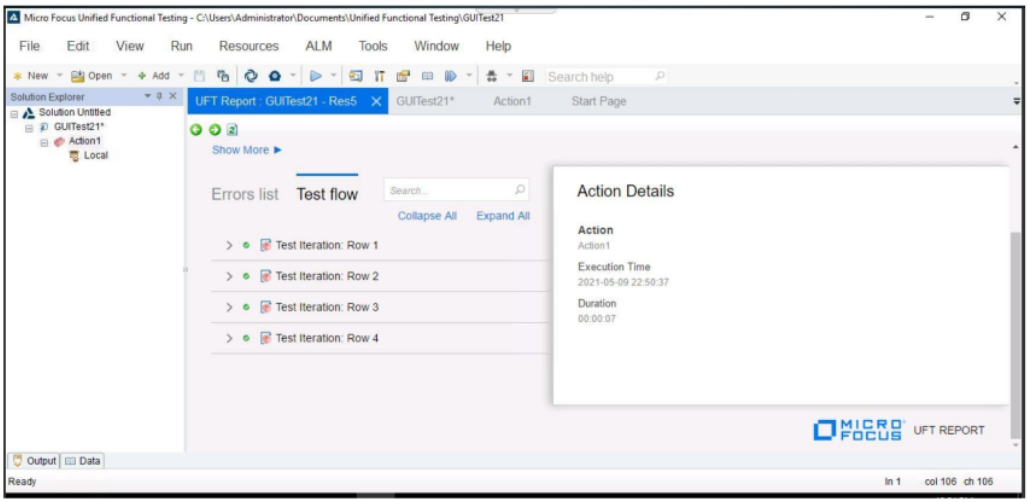
Outputs

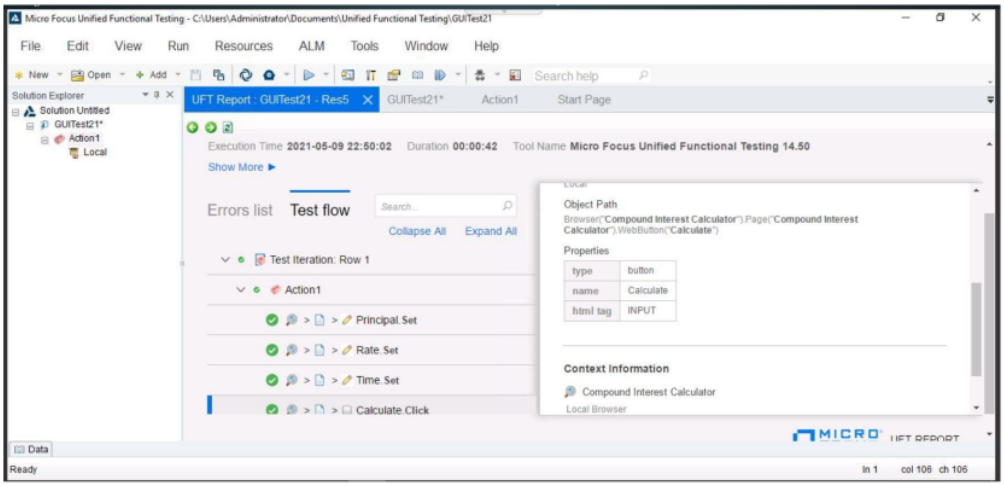



https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
2556
Rate % per Annum [R]
7
%
Time Years [n]
5
Calculate
Reset
Compound Interest [C.I.]
1028.92


https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
1322
Rate % per Annum [R]
6.5
%
Time Years [n]
6
Calculate
Reset
Compound Interest [C.I.]
606.99


https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
32.21
Rate % per Annum [R]
3.32
%
Time Years [n]
4.4
Calculate
Reset
Compound Interest [C.I.]
4.98


Micro Focus Unified Functional Testing - C:\Users\Administrator\Documents\Unified Functional Testing\GUITest21
File Edit View Run Resources ALM Tools Window Help
Solution Explorer
Solution Untitled
GUITest21*
Action1
Local
UFT Report : GUITest21 - Res5
GUITest21*
Action1
Start Page
Show More
Errors list
Test flow
Collapse All
Expand All
Test Iteration: Row 1
Test Iteration: Row 2
Test Iteration: Row 3
Test Iteration: Row 4
Action Details
Action
Action1
Execution Time
2021-05-09 22:50:37
Duration
00:00:07
MICRO FOCUS UFT REPORT
Output
Data
Ready
ln 1 col 106 ch 106


Micro Focus Unified Functional Testing - C:\Users\Administrator\Documents\Unified Functional Testing\GUITest21
File Edit View Run Resources ALM Tools Window Help
Solution Explorer
Solution Untitled
GUITest21*
Action1
Local
UFT Report : GUITest21 - Res5
GUITest21*
Action1
Start Page
Execution Time 2021-05-09 22:50:02 Duration 00:00:42 Tool Name Micro Focus Unified Functional Testing 14.50
Show More
Errors list
Test flow
Collapse All
Expand All
Test Iteration: Row 1
Action1
Principal.Set
Rate.Set
Time.Set
Calculate.Click
Object Path
Browser("Compound Interest Calculator").Page("Compound Interest Calculator").WebButton("Calculate")
Properties
type button
name Calculate
html tag INPUT
Context Information
Compound Interest Calculator
Local Browser
Data
Ready
ln 1 col 106 ch 106

CHAPTER 6
CONCLUSION

Miniature Focus' UFT One is device that utilizes robotize tests to discover issues of a test application. Brought together Functional Testing (UFT) is an abbreviation for "Bound together Functional Testing." It was recently known as QTP (Quick Test Professional).

Practical, relapse, and administration testing are the most well-known utilizations for UFT. We may utilize UFT to mechanize client exercises on a site or worker PC program, and afterward test and identify abandons for numerous clients, different informational collections, various Windows working frameworks, or potentially various gadgets utilizing similar tasks. When contrasted with manual testing, robotization by means of UFT may save a great deal of time and cash if all around planned and done. Today, UFT is quite possibly the most ordinarily used business robotized testing arrangements accessible. It's notable for its effortlessness of utilization and merchant support, just as a major local area of mechanization specialists. Accordingly, qualified UFT experts have consistently been sought after

In this manner, utilizing programming testing methods experiments were composed for the specific application. Also, with the assistance of UFT computerization instrument we're ready to test the usefulness of the site effectively.

REFERENCES

[1] Prasad, Dr. K.V.K.K. (2008) ISTQB Certification Study Guide, Wiley, ISBN 978-81-7722-711-6, p.

[2] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/software_testing_dictionary/functional_testing.htm

[3] Krishna Rungta - Founder & CEO, Feb. 2013. Accessed on: Nov. 2, 2017. [Online]. Available: https://www.guru99.com/functional-testing.html

[4] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/software_testing/software_testing_types.htm

[5] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online]. Available: https://www.tutorialspoint.com/sql/index.htm

[6] Mohammad Mohtashim - Founder & Managing Director, Feb. 2013. Accessed on: Nov. 2, 2017. [Online].Available: https://www.tutorialspoint.com/vbscript/index.htm

[7] "HP-UFT 11.50 (Unified Functional Testing)". Retrieved August 10, 2018. Available: selftechy.com.

[8] "QTP with Descriptive programming". Retrieved January 27,2011. Available: Slideshare.net.

FinalReport_171343_Neeraj_Gupta.pdf

ORIGINALITY REPORT

6%	4%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sogang University Student Paper	2%
2	www.coursehero.com Internet Source	1%
3	Submitted to Jaypee University of Information Technology Student Paper	1%
4	www.studymode.com Internet Source	1%
5	www.tutorialspoint.com Internet Source	<1%
6	www.volkantoprak.com Internet Source	<1%
7	tryqa.com Internet Source	<1%
8	Submitted to Trident University International Student Paper	<1%
9	Rajendrani Mukherjee, K. Sridhar Patnaik. "A survey on different approaches for software	<1%

test case prioritization", Journal of King Saud University - Computer and Information Sciences, 2018
Publication

10	www.guru99.com Internet Source	<1%
11	Submitted to Case Western Reserve University Student Paper	<1%
12	Submitted to iGroup Student Paper	<1%
13	cio-wiki.org Internet Source	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …17/06/21……

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | **B.Tech Project Report** | Paper

Name:_Neeraj Gupta_____**Department:**_Computer Science Engineering_____**Enrolment No:**_171343

Contact No. _7018047992_____**E-mail.** _neerajgupta651@gmail.com_____

Name of the Supervisor: _Dr. Himanshu Jindal_____

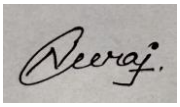
Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): __AUTOMATION TESTING -HC FACETS WITH UFT_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

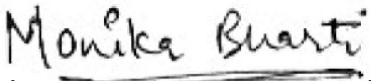
- Total No. of Pages = 64
- Total No. of Preliminary pages = 10
- Total No. of pages accommodate bibliography/references = 1



(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at …6…..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.



(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String	6%	Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com