AUTOMATION TESTING -HC FACETS WITH UFT

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

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Archita Gupta (171021)

UNDER THE GUIDANCE OF

Cognizant

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May, 2021

TABLE OF CONTENTS

DECLARATION

We 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 our work carried out under the supervision of Cognizant**.** We have not submitted this work elsewhere for any other degree or diploma.

Archita Gupta
171021

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

Cognizant
Date: 20/05/2021

Head of the Department/Project Coordinator

Project Report Undertaking

I Mr. /**Ms**. Archita Gupta, Roll No. 171021, Branch Electronics and communication Engineering is doing my internship with Cognizant from 27 February 2021 to 11 June 2021.

As per 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: Archita Gupta

Roll No.: 171021

Date: 20/05/2021

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, for her 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 tool that employs automate tests in order to find problems of a test application. Unified Functional Testing (UFT) is an acronym for "Unified Functional Testing." It was previously known as QTP (Quick Test Professional).

Functional, regression, and service testing are the most common uses for UFT. You may use UFT to automate user activities on a website or server computer program, and then test and detect defects for multiple users, various data sets, different Windows operating systems, and/or different devices using the same operations. When compared to manual testing, automation via UFT may save a lot of time and money if well designed and done.

Today, UFT is one of the most commonly utilized business automated testing solutions available. It's well-known for its simplicity of use and vendor support, as well as a big community of automation experts. As a result, qualified UFT specialists have always been in high demand

CHAPTER 1
FUNCTIONAL TESTING

Software testing is a method of evaluating program quality and lowering the risk of software error while in use. The majority of individuals have encountered software that does not perform as planned. Software that doesn't perform properly can cause a variety of issues, including financial loss, lost time, and damage to a company's brand. Death or injury

Test Execution is a part of software testing. Software testing is a multi-step process that encompasses a variety of tasks. One of these actions is execution. Software testing is a technology for finding if the program matches specified criteria and confirming that it is error free. It uses automation or manual testing for accessing many attributes of interest by working on software components. In contrast to real requirements, software testing's goal is to find mistakes, gaps, and logical errors.

1.1 Software Development Life Cycle (SDLC)

The Life Cycle for Software Development (SDLC) is a process for software design, development and testing. The SDLC offers high-quality products that satisfy client requirements and are completed on time and on schedule. The SDLC is a method for a software project inside a software firm. It is a complete technique to show how specific software may be built, managed, updated and improved. The life cycle is a technique for improving the quality of software and the development process.[1]

The many steps of a typical SDLC are depicted graphically in the diagram below

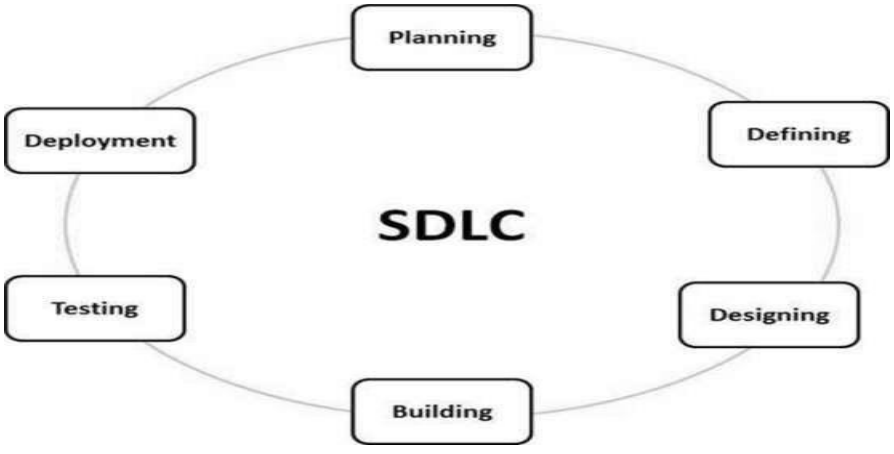


Figure1.1 Software Development Life Cycle

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

1 Planning

Analysis of requirements is the most significant and fundamental stage of the SDLC. It is done by the top members of the team with customer feedback, sales departments, market surveys and professionals from the business. These data are then used to define the core project plan and carry out product research initiatives in the economic, business and technological fields.

2 Defining Requirements

The demands of the product must be appropriately specified, documented and authorized by the customer or investment managers following the requirement analysis. This is achieved by using an SRS (Software Requirement Specifications) document that includes all project results to be established and produced through the whole project life cycle.

3 Designing the Product Architecture

For product developers wanting to build the best architecture for a new product, SRS is a resource. In a DDS – Specification of the design document, numerous alternatives to the concept suggested are usually provided and codified according to the criteria set forth in the SRS.

All key stakeholders assess this DDS and select a product design strategy based on several features such as risk management, product stability, design flexibility, budget and timescales.

4 Building or Developing the Product

At this phase of the SDLC the actual development of the product begins. The programming code is produced in accordance with DDS at this phase. When the design is done accurately and structural, code generation may be done swiftly and easily.

5 Testing the Product

Since testing in modern SDLC designs is primarily incorporated in all phases of SDLC, this stage is often an all-phase subset. This phase nevertheless concerns just the testing phase of the product, during which problems are reported, monitored, fixed and checked until the product fulfils the SRS quality requirements.[2]

6 Deployment in the Market and Maintenance

Once the product is entirely tested and ready to deploy, it is officially published on the relevant market. Depending on the business model of the firm, product roll-out often takes place in stages. Depending on input, the product can then be released as or with suggested improvements in the market group. It is maintained for the present customer base after the products have been released to the market. [1]

1.2 Software Testing Objectives

Figure1.2 Testing Objectives

To evaluate the work products

Before the developer picks up the work items for development, such as the Requirements specification, Designs, and User Stories, they should be checked. Recognizing any uncertainty or conflicting requirements at this point saves a lot of time during development and testing. Before the code integrates/is ready for testing, it undergoes static analysis (reviews, walk-throughs, inspections, and so on). Verification is the term for this type of examination. During the development phase of any work product, it is the process of reviewing the product.

To verify requirements:

This goal demonstrates that one of the most important aspects of testing is to meet the demands of the client. Testers examine the product and ensure that all of the stipulated standards are met. Designing every test cases, irrespective of testing approach, guarantees that every performed test case is functionally verified. A requirement traceability matrix (RTM) should also be created by the tester

to guarantee that all test cases are mapped to requirements. RTM is a powerful tool for ensuring that test cases cover all of the requirements.

To validate the test object:

Testing guarantees the execution of requirements as well as the assurance that they function as intended by users. Validation is the term for this type of testing. It is the process of inspecting a product after it has been developed. Validation might be carried out manually or automatically. It frequently uses a variety of testing approaches, such as Black Box, White Box, and so on. Validation is often performed by testers, although customers can also validate the product as part of User acceptability testing. The consumer is regarded as the king in any firm.

To build confidence:

Improved software quality is one of the most important goals of software testing. There are fewer flaws in high-quality software. To put it another way, the more effective the testing process is, the less faults will be found in the final output. As a result, the overall quality of the test item will improve. Excellent quality leads to higher customer satisfaction and fewer maintenance costs.

To prevent defects:

One of the goals of software testing is to avoid problems early in the development process. The expense and effort of detecting problems early saves a lot of money and time. Defect prevention entails doing a root cause of previously discovered problems and then taking specific steps to avoid similar errors from occurring in the future. Effective testing aids helps in the creation of an error-free product. When faults are prevented, the overall defect count in the product is reduced, resulting in a higher-quality product for the consumer.

To find defects in product:

Another important goal of software testing is to find all flaws in a product. The goal of testing is to uncover as many flaws as possible in a software product while also ensuring that the application meets the user's needs. Defects should be found as early as feasible in the testing cycle.

To share information to stakeholders:

The goal of testing is to give stakeholders all of the information they need concerning technical or other constraints, risk factors, confusing requirements, and so on. It might take the form of testing activities or testing results that highlight what's missing and what was wrong.

To reduce the level of risk:

Risk is a word which means to the chance from losing. Goal of this testing is for lowering the chance of a risk occurring. If we don't manage these uncertainties, possible hazards will arise not just during the development period, but also throughout the product's whole life cycle. [2]

1.3 Benefits of Software Testing

• Cost-Effectiveness: This is the most significant benefits of software testing. Testing on time of any project allows us to save cost in the future. It is less expensive to correct issues discovered early in the software testing process.

• Security: Software testing's most susceptible and sensitive advantage is security. People are seeking for items that they can trust. It aids in the early detection of hazards and issues.

• Product quality: Any software product must meet this criterion. Testing guarantees that buyers receive a high-quality product.

• Customer Satisfaction: The primary goal of every product is to provide customer satisfaction. The optimal user experience is ensured via UI/UX testing.[2]

1.4 Test Process in Software Testing

Testing is more of a procedure than a single task. Testing must be planned, and following through on it necessitates discipline. The quality of the test methodology employed determines the quality and efficacy of software testing. The testing process may be broken down into the following fundamental steps:

1) Planning and Control

Test Planning : Test planning involves the production of a document outlining a broad approach and testing objectives. This process includes the assessment of the test foundation, the selection of the test conditions to analyze the tests, the development of test cases and the design of the test environment. To be sure, completion or exit conditions have to be set while testing (at any phase).

Purpose

• To assess the scope and hazards of testing, as well as the testing objectives.

• To determine the necessary test resources, such as personnel and test environments.

• To schedule tasks such as test analysis and design, as well as test execution, execution, and assessment.

2) **Analysis and Design**

• The following are the key tasks of test analysis and test design:

• To go through the testing basis again. The test base is referred to as knowledge concerning test cases such as needs, thorough design, product risk analysis, architecture and interfaces.

• Test circumstances to be determined

• Tests are created

• Planning the test environment and identifying the technologies and tools necessary.[3]

3) **Implementation and Execution**

Execution of test entails performing a required test on a computing device, manually or else with the help of an automation testing tool.
The following is the main task of test implementation:
• Using methodologies, construct and prioritize test cases, as well as collect test data for those tests.
• Also, for confirming a repair, run the tests once again that had shown status as fail
• Also, for keeping track of the results of the test execution. The state of the test case is recorded in a test log.

4) Evaluating Exit criteria and Reporting

Exit criteria are used to determine when testing should be stopped. It is determined by code coverage, functionality, and risk. In general, it is affected by market risks, cost, and timeliness, and varies from project to project. When the following conditions are met, exit criteria are used:
• A particular percentage of test cases are completed with a specific pass rate.
• The bug rate falls below a particular threshold
• When we meet our deadlines

5) Test Closure activities:

When the software is ready to be deployed, test closure operations are completed. Testing can be terminated for a variety of reasons, including:
• the cancellation of a project
• the achievement of a certain goal
• When there is a maintenance release or an update

The following are the key duties of test closure activities:
• Verify that all scheduled deliverables have been provided and that each and every functionality in reports have been addressed.
• Handing over the test equipment to the maintenance team. They will assist with the software.
• To assess the testing process and draw lessons for future releases and projects.[3]

1.5 Levels of Testing

1. **Unit testing:** determines whether software components perform as expected.

2. **Integration Testing**: examines the flow of data from one module to another.
3. **System Testing**: assesses both types that is functional and non-functional testing .
4. **Acceptance Testing**: Verifies that a specification or contract's criteria are satisfied in accordance with its delivery.

Each of these layers of testing has a distinct purpose. The software development lifecycle benefits from these levels of testing.

1) **Unit testing:** Unit is the individual component of a system or program that can be built, loaded, and run. This type of testing allows each module to be tested independently.
The goal is to test each component of the software separately. It determines whether or not the components are performing their functions. Developers are the ones who do this sort of testing.

2) **Integration testing:** The term "integration" refers to the process of merging two or more , like separate software modules are joined and after that they are tested in a group in this testing step to ensure that the connected system is ready for testing phase. Testers are the ones that do this type of testing.

3) **System testing:** This is done on a functional, integrated system. It enables for the verification of the system's compliance with the standards. It examines the overall interplay of the various components. Load, efficiency, stability, and security testing are all part of the process. System testing is frequently the last step in the process of ensuring that the system fits the requirements. It assesses both functional and non-functional testing requirements.

4) **Acceptance testing:**

This type of testing uses proper steps for determining if the criteria of a specification or contract have been satisfied as of the date of delivery. The user or customer is the one who does acceptance testing. Other stockholders, on the other hand, may be active in the process.

Types of acceptance testing:

Alpha testing

This is another type of testing that looks for software's defects and flaws. This type of test is carried out towards the end of the application development, before the product is launched or delivered to

the customer, to verify that the user or client receives an error-free software program. Alpha testing comes before beta testing, therefore you'll need to run beta testing once you've completed alpha testing.
Alpha testing does not take place in a real-world setting. Rather, these assessments are carried out in a virtual environment that closely resembles the actual world.

Beta testing

Beta testing follows alpha testing, as previously stated. Before a product is released, it is subjected to beta testing. It is carried out in a real-world context by a small number of registered users and customers in order to ensure that the program is error-free and runs properly. Some improvements are performed to improve the program after collecting input and constructive criticism from those users. As a result, while software is in beta testing, it is referred to as a beta version. After this round of testing is completed, the program is made available to the general public.[4]

1.6 Types of testing

1.6.1. FUNCTIONAL TESTING

It is a sort of software testing in which the software system is validated against the functional specification or requirements. Functional tests are used to validate the output of a software program by giving adequate input and comparing it to the functional requirements. [4]

Objective of this testing

• **Fundamental Usability**: It entails basic usability testing of the system.
• **Mainline Functionality**: It entails testing the primary functions of an application. It helps the tester to find if a user could browse across the screens totally or without any issue.
• **Accessibility:** It confirms that the system is easily accessible to the user.
• **Error Conditions**: Error conditions are checked using testing methodologies. It examines if the appropriate error messages are presented.

1.6.2 NON-FUNCTIONAL TESTING

It is a type of software test that investigates the non-functional elements of a software program (performance, usability, dependability, and so on).

Objectives of this testing

• Nonfunctional testing should help to improve the program usability, efficiency, maintainability, and portability.
• Assists in lowering a risk or expense of non-functional parts of the product throughout production.
• Improve the method in which the product is downloaded, configured, run and maintained
• Gain a better understanding of how products behave and what technologies are in use.

Characteristics of Non-functional testing

• This testing must be quantitative, thus subjective values like good, better, best, and so on must not be used.
• Exact figures are unlikely to be found at the outset of the requirement process.
• Prioritizing each and every need is critical.
• Ascertain that quality attributes are accurately recognized in Software Engineering.

1.6.3 Black Box Testing

Black Box Testing is an evaluating software methodology which includes testing software applications functionalities without understanding the code structure, implementation details or internal pathways. Black Box Testing is a sort of software testing which focuses on software products outputs and inputs and is fully driven by software specifications. Another name for it is behavioral testing.

Figure1.3 Black Box testing

Black Box Testing Techniques

• **Equivalence Class Partition**: This technique is intended to keep the amount of feasible test cases to a minimum while yet ensuring adequate coverage.
• **Boundary Value Analysis**: BVA also checks the values at the edges. This technique assesses if the range of values is acceptable to the system. It's a great way to cut down on the amount of test cases. It's best for systems with inputs that fall between specific ranges.
• **Decision Table**: A decision table is a matrix that connects causes and consequences. Each column has a different combination.

1.6.4 White Box Testing

This is another type of testing which is a software review method that comprises testing the underlying structure, architecture and code of the product in order to evaluate the flow of input and improve the design, usability and safety. Clear box testing, open box testing, transparent box testing, code-based testing, and glass box testing is sometimes known as white box testing since the code is visible to testers.

Because of the see-through box concept, the nickname "White Box" was coined. The term "clear box" or "White Box" refers to the ability to look into the software's inner workings via its exterior shell (or "box"). Similarly, the "black box" in "Black Box Testing" denotes the inability to observe the software's inner workings, allowing only the end-user experience to be assessed.

Verification of this testing

•Vulnerabilities in internal security
•Breaking or badly designed coding paths
• The flow of certain inputs via the code
• Expected output
• Conditional loop functionality
•Each statement, object, and function must be tested separately.

White Box Testing Techniques

Code Coverage Analysis is a popular White box testing approach. A Test Case suite's holes are filled through Code Coverage analysis. It detects parts of a program that aren't put to the test in a collection of test cases.
Code coverage analysis may be performed using automated technologies. A box tester can employ the following coverage analysis techniques:

Statement Coverage: During the testing of code , this methodology mandates that every conceivable statement present in the code must be tested at least once.

Branch Coverage - This methodology examines every conceivable path of a software application (if-else and other conditional loops).

Advantages of White Box Testing

• Optimizing the code by detecting hidden faults.

• Test scenarios for white boxes are easy to automate.

• Testing is usually more extensive since every code path is covered.

• Tests may start early on in the SDLC even if a GUI is not available.

Disadvantages of White Box Testing

• Testing the white box might take time and money.

• Developers used to test scenarios of white box detest it. The lack of information from developers in the testing of white cases might lead to difficulties in production.

• Professional resources are required for white box testing with a solid understanding of programming and implementation.

• Testing white box takes time; bigger application programming needs more time for complete testing.

1.6.5 Dynamic Testing

Dynamic testing is a type of software testing that examines the dynamic behavior of software. The basic goal of dynamic testing is to uncover flaws in the program runtime environment by evaluating software behavior with dynamic variables or variables that are not constant. In order to test the dynamic behavior, the code must be run.

Dynamic testing Goals

The primary goal of this type of testing is to verify if the software functions accurately both before and after installing it, resulting in a reliable program free of severe problems. The major goal of the dynamic test is to guarantee that the program is consistent. Consistency relates to a variety of requirements such as performance, usability, combativity, and so on, making Dynamic Testing extremely crucial.[4]

Dynamic Testing Advantages

• This type of testing could disclose undiscovered flaws that are too tough or complex to be captured by static testing.

•In this type of testing, we run the program from beginning till the end, assuring code without error and therefore improving the performance of the software.

• This type of testing has become indispensable for spotting security threats.

Disadvantages of Dynamic Testing

• This type of testing takes huge amount of time since it runs the application/software or code, that consumes a lot of assets.

• Because this type of testing do not begin early in the lifecycle of software, any errors that are corrected later in the process might raise the cost of the project/product.

1.6.6 Static testing

This type of testing is a software testing technique that looks for errors without having to run the code in the program. It is mainly used in catching faults in the starting of development process, when they are easier to spot and correct. It also helps the tester in the detection of faults that Dynamic Testing may not be able to detect. Dynamic Testing, on the other hand, examines an application while the code is running. These are basic procedures of static testing:

• **Testing program manually:** Tests done manually, often called as reviews, include manual code analysis.

•**Tool-assisted automated analysis:** Tool-assisted automated analysis is essentially static analysis.

Execution of Static Testing

1. **Validation of Use Cases Requirements**: This ensures that all end-user activities, as well as any input/ output, are identified. The test cases can be more accurate and complete if the use cases are more explicit and complete.

2. **Validation of Functional Requirements**: This ensures that all relevant aspects are identified in the Functional Requirements.

3. **Architecture Review:** This includes all business-level processes such as server locations, network diagrams, protocol specifications, load balancing, database accessibility, test equipment, and so on.

4. **Validation of Prototypes/Screen Mockups**: This step comprises the verification of and use cases.

1.6.7 Confirmation testing

Confirmation testing is a sort of software testing in which testers retest a software product to ensure that previously reported defects have been repaired or are no longer present. When a test fails, testers usually report a defect. Testing team will then retest to see if the reported problem has been addressed. Confirmation testing is the term for this. Re-testing is another term for confirmation testing. In order to duplicate the problem, testers must refer to the defect report that was prepared when the problem was first reported. The defect report assists the tester in performing the test by ensuring that the same test processes, test data, and test environment are used.

Execution of Confirmation Testing

Confirmation testing is carried out under the following situations within the software testing life cycle.

1. Whenever the development team releases a new build with bug fixes
2. Verify the bug fix before doing regression testing.
3. When the development team rejects a problem report from a tester. To replicate the problem, testers do confirmation testing.

1.6.8 Regression testing

It is a sort of software testing used for ensuring that a program or code modification hasn't broken current functionalities. Regression Testing is just a complete or partial re-execution of previously performed test cases to confirm that current functionality is working properly. This testing ensures that new code modifications do not have unintended consequences for current functionality. It guarantees that the old code continues to function after the most recent code modifications have been made.[4]

Need of Regression Testing

This type of testing emerges most often when there is a need to alter the given program and also if we want to see if the changed program impacts different parts in the given program. It is also required when a developer adds or modifies feature of a program, as well as for bug and performance issues.

Regression Testing Tools

Regression testing expenses will rise if your software is often updated. Testing the application manually increases the time of execution of test and expenses in such circumstances. In such

instances, automating the software using this tools is the best option. The amount of automation is determined by the number of test cases that may be reused in subsequent regression cycles.

1.Selenium: It is the opensource tool for web application automation. Selenium is a browser-based regression testing tool.

2.**Quick Test Professional (QTP)**: QTP is an automated software that may be used to automate functional and regression test scenarios. This automates using VBScript programming. It's a keyword-driven, data-driven tool.

3.Rational Functional Tester (RFT): This is a Java tool for automating software application test scenarios. It interfaces with Rational Test Manager and is mostly used for automating regression test scenarios.

Challenges in Regression Testing:

The most common regression issues:

• Test suites grow in size as more regression runs are performed. The whole regression test suite cannot be run due to time and financial restrictions.

• Reducing the test scenarios while maintaining maximum amount of coverage is still the problem.

• Determining the frequency of these Tests, such as later each alteration, build update, or set of issue repairs, is difficult.[4]

1.6.9 Smoke Testing

This is a software testing method which assesses whether a system package is delivered and is maintaining stability. "Build Verification Testing" or "Confidence Testing" are other terms for smoke testing. To put it another way, we're checking to see whether the main features are working and if there are any show-stoppers in the build we're testing.

This allows you to see if the build is faulty, preventing you from wasting time and money on further testing. The smoke tests indicate that the structure is ready for further formal testing. The primary goal of smoke testing is to uncover severe problems early on. Smoke tests are used to show that a system is stable and meets the criteria. All data files, libraries, reusable modules, and engineering components necessary to accomplish one or more product functionalities are included in a build.

Execution of this testing

The development team deploys the build in QA in this testing procedure. Testers perform test cases on the build after taking selections of test cases. The application's essential functionality is tested by the QA team. The purpose of this set of test cases is to expose construction errors. If these tests pass, the QA team will move on to Functional Testing.

Need of smoke testing

Smoke testing is vital in software development since it assures the system's accuracy in the early stages. We can save time and money by doing so. As a consequence, smoke tests ensure that the system is in proper working order. Only when we've completed smoke testing will we begin functional testing.

• Smoke testing will identify all of the build's show stoppers, and it will be done after the build has been released to QA. Smoke testing identifies the majority of errors during the early phases of software development.

•Smoke testing makes detecting and correcting severe flaws much easier.

• Smoke testing allows the QA team to identify application functionality flaws that may have developed as a result of the new code.

• Smoke testing identifies the most serious errors.

Smoke testing cycle

The flow chart below depicts how Smoke Testing is carried out. We move on to functional testing when the build has been given to QA and the smoke tests have passed. If the smoke test fails, we stop the test and wait for the build problem to be resolved.

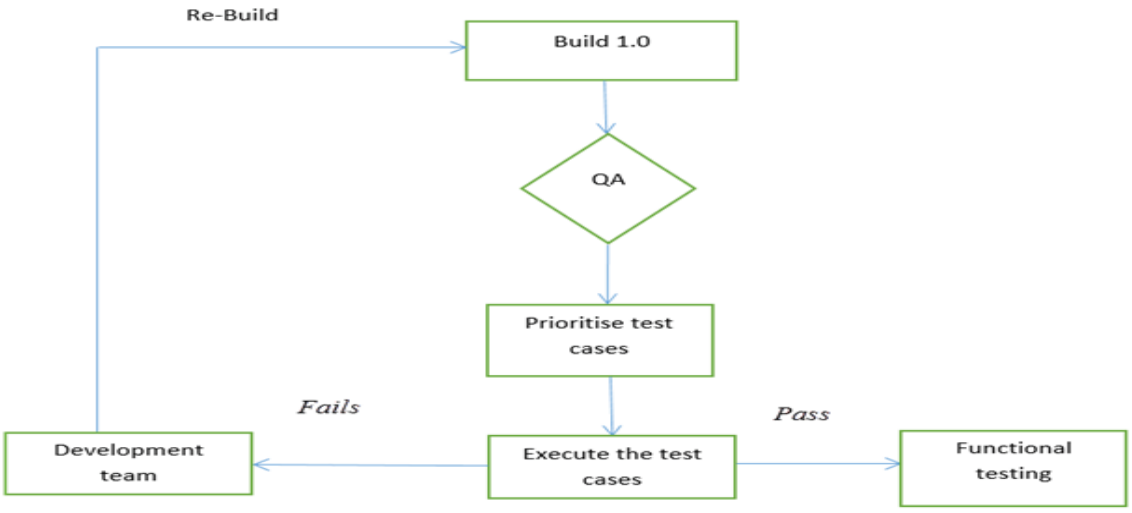


Figure1.4 Smoke test cycle

1.7 Test case writing

It is a sequence of steps performed for ensuring that the certain functionality or capability of our software program is working properly. Test Case means a collection of test procedures, data, preconditions, and postconditions created for a particular test scenario in order for validating any requirement. The test case contains specified circumstances that a tester might use to tally between expected results and actual results in order for assessing if the application meets the customer's needs.

1.7.1 Test Case Management Tools

These are automation solutions which assist in the administration and upkeep of Test Cases.

1. **Documentation of Test Cases**: You may utilize tools to speed up the construction of Test Cases by using templates.
2. **Run the test case and keep track of the results**: The tools may be used for running a test , and the outputs may be conveniently documented.
3. **Automate Defect Tracking:** The tests which do not pass are immediately connected to a bug tracker, afterwards it can be allocated to coding team and followed up on through email alerts.
4. **Traceability**: Requirements, Test cases, and Test case execution are all connected by the tools, also every test case can be compared to the previous one to evaluate how many test cases are covered.

1.7.2 Benefits of Writing Test Cases

The main goal of a test case is to guarantee that various functionalities of an application perform as planned. It assists testers in determining if the program is defect-free and meets the requirements of end users. Other advantages of test cases include:

ensuring good test coverage,

helping to enhance software quality, and so on.

Reduces the cost of maintenance and software support

Assist in ensuring that the program satisfies the needs of the end user.

Allows the tester to think deeply about the tests and approach them from as many perspectives as possible.

Test cases are reusable in the future since they may be referenced and executed by anybody.

These are a few of the reasons why test cases are so important in software testing. Test cases are strong artefacts that serve as a reliable source of information about how a system or a specific feature of software operates.

1.7.3 Test Case Format

An ID, description, a set of inputs, a few actionable actions, and expected and actual outcomes are the main components of a test case.

Test Case Name: A test case should have a self-explanatory name or title.

Test Case Description: In a few words, the description should describe the tester what they'll be testing.

Pre-Conditions: Include any assumptions that apply to the test as well as any preconditions that must be satisfied before the test can be run.

Test Case Steps: The test steps should include all of the essential data as well as instructions for running the test. The stages should be clear and concise, without excluding important information.

Test Data: It's critical to choose a data collection that provides adequate coverage. Choose a data collection that includes not just good but also negative possibilities.

Expected Result: The expected outcomes describe what the tester should anticipate to see as a result of the test procedures.

Actual Result: They describe how the application performed during the execution of test cases.

Comments: Any additional information, such as screenshots, that the tester wishes to share might be added here.

This is the standard format used by testers when writing test cases. Additional parameters, such as test case priority, kind of test case, and bug severity, can be added to these parameters by testers.

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

1.8 Bug Report in Software Testing

In software testing, a bug report is a thorough document describing the flaws detected in the software program. A bug report comprises every detail regarding a bug, such as the description, the date the bug was discovered, the identity of the tester who discovered it, the name of the developer who corrected it, and so on. A bug report aids in the detection of similar issues in the future, allowing them to be avoided.

The following information should be included in the Problem Report when reporting the bug to the developer.

• **Defect ID** - The defect's unique identifying number.

• **Defect Description**- A detailed description of the Defect, including details on the module where the Defect was discovered.

• **Version** - The program version in which the flaw was discovered.

• **Steps** - a detailed set of steps with screenshots that the developer may use to replicate the flaws.

Date Raised - Date when the defect is raised

• **Reference**- where you provide references to papers like as specifications, design, architecture, or even images of the problem to aid comprehend the fault

• **Detected By** - The name/ID of the tester who reported the flaw.

Status - Status of the defect

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 developer who resolved it

• **Date Closed** - the date when the fault was fixed

• **Severity** - the severity of the fault's impact on the application

• **Priority**, which is tied to the urgency of resolving defects. According to the effect urgency with which the issue should be rectified, the severity priority might be High, Medium, or Low.[4]

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

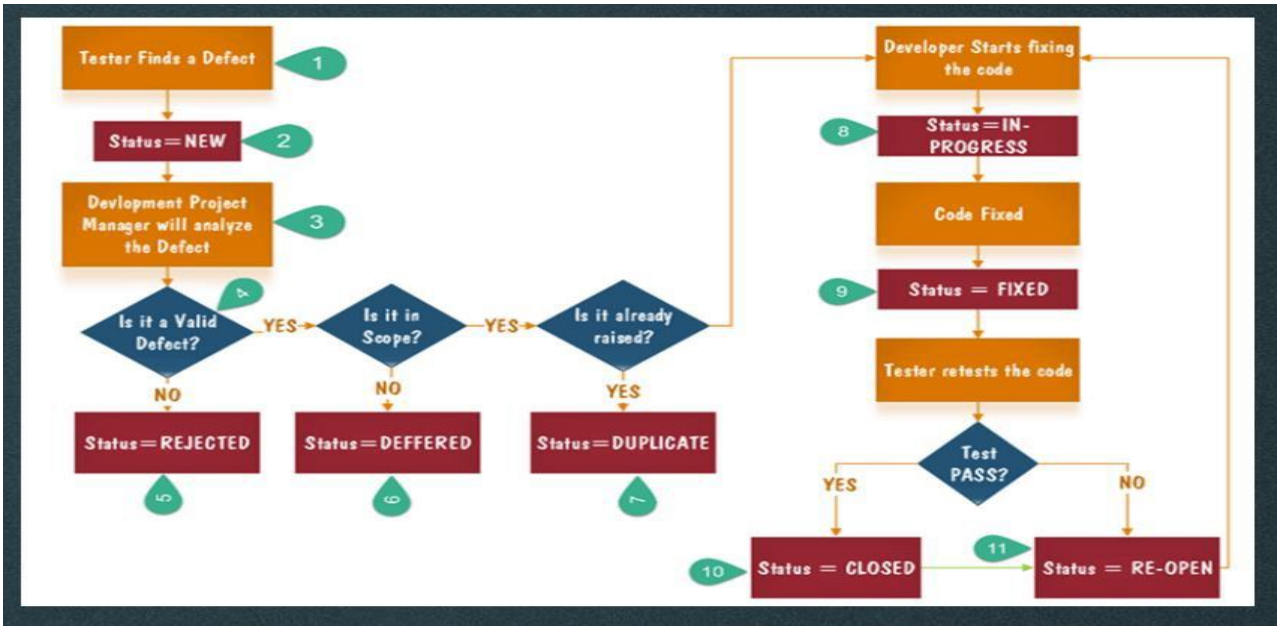


Figure1.5 Bug Life Cycle

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 previously said, SQL is one of the most extensively used database query languages.

• Provides access to data stored in relational database management systems to users.
• Allows users to annotate database.
• SQL modules, libraries, and pre-compilers may be used to embed in other languages, allowing users to define and modify data in a database. Users may create and remove databases and tables with this program.
• Provides users with the ability to create table queries, database objects, and operations.
• Users can change the restrictions for columns, functions, and objects.

2.1.2 SQL Process

If we perform a Sql query towards any RDBMS, our software selects the optimal approach to fulfil given request and the SQL engine determines when to comprehend it. There are several components to this operation. These are the elements:
• Engines for optimization
• Dispatcher of Query
• Search Engine (Classic)
• SQL Query Engine, for example.
A traditional query engine handles non-SQL queries, whereas a SQL query engine handles logical files. The SQL Architecture is depicted in the figure below.[5]

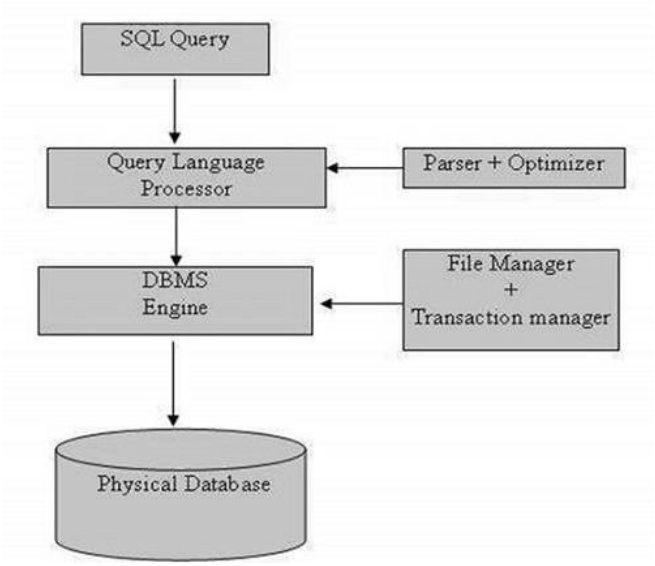


Figure2.1 SQL Architecture

2.1.3 RDBMS

"Relational Database Management System" is an acronym for "Rdbms." A relational database management system, or RDBMS, is a database management system created exclusively for relational databases. RDBMS are thus a subset of DBMS.

It is one that uses rows and columns to store data in an organized way. This makes finding and accessing certain values inside the database a breeze. As its values inside each field are correlated to one another, it is called "relational." Tables can also be linked to one another. Because of the relational nature, queries may be conducted on several tables at the same time.

2.2 XML

The abbreviation for (XML) stands for "Extensible Markup Language." Despite Html elements, which specify how the data should be shown, XML tags analyze the information and can be used for storing and maintaining it rather than specifying how it should be shown. In the near future, XML would not eliminate HTML, but this will access more information by combining many of HTML's excellent characteristics.

Three essential characteristics of XML make it useful in a wide range of systems and solutions:

• **XML is extensible**: We may design our tags giving our own description and in any language, to fit our software.

XML transfers data but will not show it. Why does XML carry data but not show it? You can save data in XML irrespective of how it might be displayed.[5]

• **XML is a freely accessible standard**. The World Wide Web Consortium (W3C) developed XML, that is free and open technology.

2.2.1 Usage

XML could operate behind at scenes to ease the generation of HTML content for huge web sites, according to a brief list of its uses.
Information may be sent between organizations and companies using this.
XML may be used to unload and reload databases.
Data may be stored and organized using XML, makes it easy to customize particular relational database needs.
XML plus style sheets may be easily coupled to provide almost any correct outcome.
Almost any type of data may be sent using an XML document.

2.2.2 Markup Description

XML is a markup language for describing of rules for classify observations in a way that is both physical and digital readable. Markup is information that is appended to something like a document to improve its meaning in certain ways, such as identifying the elements and their interactions. It is a set of symbols that may be put into a document's text to distinguish and label its different portions.

2.3 JSON

JSON, or JavaScript Object Notation, is a text-based open standard for transferring human-readable data. Programmers are familiar with JSON conventions and other programming languages.
• JSON format was created by Douglas Crockford
• It was created to exchange data in a human-readable format.
• The JavaScript programming language has been expanded.
• The .json file extension is used.
• JSON is a type of data format. Application/json is the Internet Media Type.
• The Uniform Type Identifier (UTI) is open to the public.
json is a kind of data.[5]

2.3.1 Uses of JSON

• It's utilized in the development of applications which are java based, such as extensions of browser and webpages.
• It is a serialization or transmission format for data which is structured over a connection of networks.

• This is mostly used for send information from a server to website apps.
• JSON is the format used by web based services and APIs to expose data which is not private.
• This is compatible by today's programming languages.

2.3.2 JSON Characteristics

• It is simple in understanding and writing.
• This is an interchangeable format which is based on text and that's easy to use.
• JSON is language agnostic.[5]

CHAPTER 3

VBSCRIPT

Visual Basic Scripting, or VBScript, is a subset of Visual Basic for Applications (VBA). This is a Microsoft product that may be found not just in Microsoft products like MS Project and MS Office, and can be used in different programs like AUTO CAD.

3.1 VBScript features

• It's a simple scripting language with a lightning-fast processor.

• VBScript is case-insensitive for the most part. It has a very simple syntax that is easy to pick up and utilize.

•VBScript is not an Object-Oriented Programming language like C++ or Java, but rather an object-based scripting language.

• To access the aspects of the environment in which it is running, it employs the Component Object Model (COM).

• Only in a Host Environment, such as Internet Explorer (IE), Internet Information Services (IIS), or Windows Scripting Host, can VBScript be successfully run (WSH)

3.2 VBscript – Version History and Uses

Microsoft released the first version of VBScript in 1996, and it was version one. Most recent stable version of VBScript is 5.8, which is included with Internet Explorer 8 and Windows 7. The places where VBScript may be used are numerous and are not limited to the ones listed below.

• Quick Test Professional, abbreviated as QTP, is a popular automation testing application that uses VBScript as a scripting language.

• Windows Scripting Host, which is mostly used for automating the Windows Desktop by Windows System Administrators.

• Active Server Pages is a scripting environment which has server that employs VBScript or Java Script to create dynamic webpages.

• In Microsoft Internet Explorer, VBScript is utilized for client-side scripting.

•VBScript is used to execute Microsoft Outlook Forms, whereas VBA is used for application- level programming (Outlook 2000 onwards).[6]

Disadvantages

• Only Internet Explorer browsers support VBscript. VBScript is not supported by other browsers, such as Chrome and Firefox. As a result, JavaScript takes precedence over VBScript.

• VBScript's command-line functionality is limited.

•Debugging is tough due to the lack of a default development environment.

VBScript usage

The latest version of VBScript is 5.8, also with the latest introduction of the.NET framework, Microsoft has opted to continue to support it for web development within ASP.NET. As a result, no new versions of the VBScript engine will be released, however the Microsoft sustaining Engineering Team will handle any defect fixes and security problems. The VBScript engine, on the other hand, would be included by default in all Microsoft Windows and IIS installations.[6]

CHAPTER 4
UFT AUTOMATION

Unified Functional Testing, or UFT, is an automated functional testing tool that assists testers in executing automated tests in order to find any mistakes, faults, and spaces within application under test that are not consistent with the expected results. Mercury Interactive created it, which was eventually bought by HP and is now Microfocus. QTP stands for Quick Test Professional, while UFT stands for Unified Functional Testing.

UFT, originally Quick Test Professional (QTP), is a piece of software that automates functional and regression testing for software applications and environments.

It has a graphical user interface and supports keyword and scripting interfaces. It specifies a test method and manipulates the objects and controls of the application under test using the Visual Basic Scripting Edition (VBScript) scripting language. From a single console, developers may test all three layers of a program's operations: the interface, the service layer, and the database layer.

Quick Test Professional was the initial name for UFT, which was created by Mercury Interactive. Hewlett-Packard (HP) eventually bought Mercury Interactive in 2006. Up until 2016, when the HP Software Division was sold to Micro Focus, UFT 11.5 integrated HP Quick Test Professional and HP Service Test into a single software package that was accessible through the HP Software Division.

4.1 DESCRIPTION

This is a type of software that automates the tests of a variety of software applications and settings. It uses a user interface, such as a native GUI or a web interface, to do functional and regression testing. It operates by recognizing objects in an application's user interface or a web page and performs specified activities; it may also record object data such as property and id of handler. To describe the test method and modify the objects and controls of the application under test, This software uses the VBScript scripting language. Users may need to alter the underlying to execute more complex activities.[7]

Exception handling

Micro Focus UFT handles exceptions via recovery scenarios, with the purpose of continuing to execute tests even if there is an unexpected failure. Because UFT connects further into system memory of the programs being tested, various events may cause HPE Unified Functional Testing to abort and be unrepairable.

Data-driven testing

Focus on the small details Data-driven testing is encouraged at UFT. Information could be exported to a excel sheet, for example, and reused somewhere else. This is carried out using a Microsoft Excel worksheet which may be approached using UFT. The Global data sheet and the Action (local) data sheet are the two types of data tables used by UFT. The test stages can take data from these data tables and drive variable data into the application under test, ensuring that an anticipated outcome is achieved.

Automating custom and complex UI objects

Focusing on the small details Customized user interface items and other complicated things may be missed by UFT. These things can be classified as virtual objects or insight objects by users. Virtual objects are hardly supported by UFT for analogue recording or recording in low-level mode.

Extensibility

This may be customized with add-ons having a wide range of development environments which aren't carried outside the app. Web, .NET, Java, and Delphi are all supported by UFT add-ins. HP Functional Testing software includes HP Quick Test Professional and the HP Quick Test Professional add-ins.

User-Interface Design

It gives two views of a test script, as well as options to alter it: Keyword View and Expert View. These views enable UFT to operate as a test's Integrated Development Environment (IDE), and UFT contains several conventional IDE capabilities, such as breakpoints, which allow you to halt a test at certain points.

Keyword view

Users may utilize Keyword View to build or check test procedure in a modular, tabular manner. Each row indicates a step that can be changed. Any of the following columns can be included in the Keyword View: Item, Operation, Value, Assignment, Comment, and Documentation. UFT shows a corresponding line of script for each stage in the Keyword View based on the value of row and column . At any time, users can add, delete, or alter steps.[7]

Users may also inspect attributes for objects like checkpoints, output values, and actions in Keyword View, as well as utilize conditional and loop expressions and insert breakpoints to help debug a test.

Expert view

It allows people to view or change the source code of a test using VBScript. Testers can alter all test steps except the root Global action, and modifications are synced with the keyword view, which is designed for more experienced users.

Languages

The scripting language used by Micro Focus UFT is VBScript. Classes are supported by VBScript, however polymorphism and inheritance are not. VBScript loses the ability to utilize some Visual Basic keywords, as well as an integrated debugger, an event handler, and a forms editor, as compared to Visual Basic for Applications (VBA). Although HP has included a debugger, it is restricted in comparison to testing tools that include a full-featured IDE, such as those included with other programming languages.[8]

4.2 Drawbacks

Micro Focus UFT is a Windows-based application. It is based on somewhat outmoded Windows-only technology like ActiveX and VBScript, neither of which is an object-oriented language. This software unable to test with all browser versions and kinds. It does not, for example, support Opera.

Although remote execution is still available with HPE Unified Functional Testing running on a separate system, the Test Execution engine is coupled with the GUI Test Code development IDE, therefore there is no way to run the tests independently of UFT.

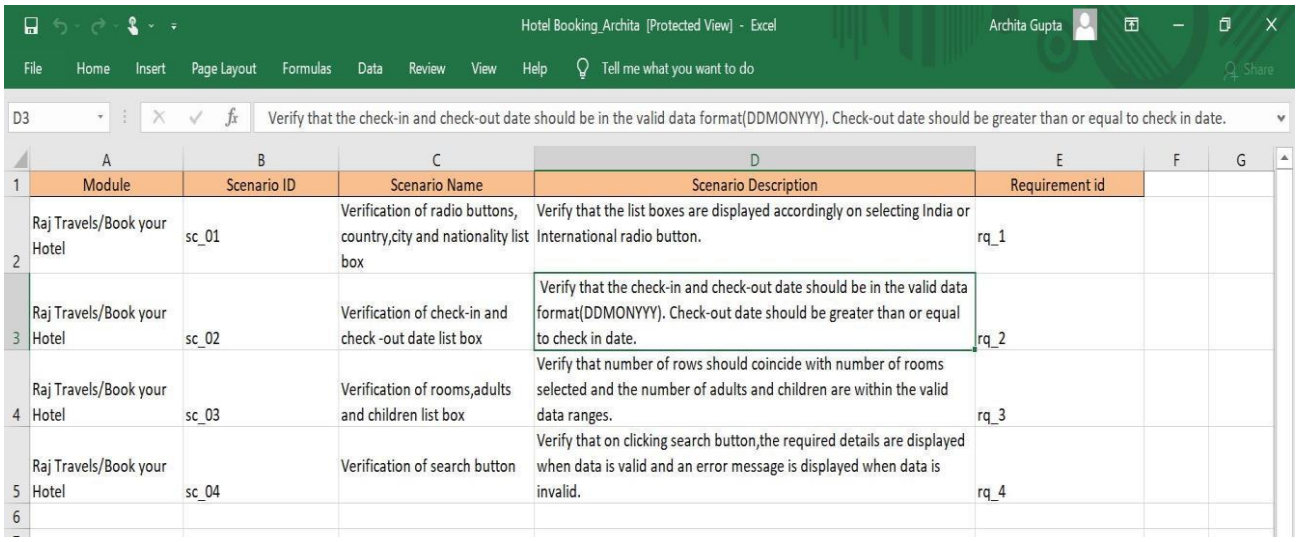
Because of the high license fees, the technology is frequently only utilized by a small group of testers inside a company. This supports a compartmentalized approach to QA/testing, where test is done separately from the business and development teams, rather than a collaborative approach where testers collaborate closely with the developers and business team.[8]

CHAPTER 5

DESCRIPTION OF WORK CARRIED OUT

5.1 Writing test cases for hotel booking website

Test Scenario



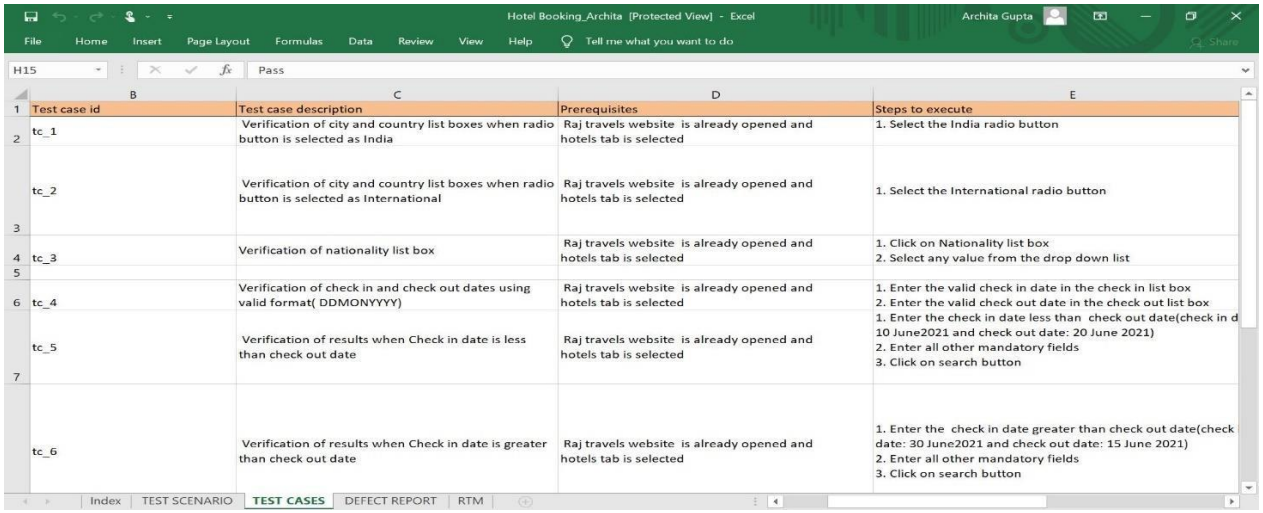
Hotel Booking_Archita [Protected View] - Excel — Archita Gupta

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

D3: Verify that the check-in and check-out date should be in the valid data format(DDMONYYY). Check-out date should be greater than or equal to check in date.

	A	B	C	D	E	F	G
1	Module	Scenario ID	Scenario Name	Scenario Description	Requirement id		
2	Raj Travels/Book your Hotel	sc_01	Verification of radio buttons, country,city and nationality list box	Verify that the list boxes are displayed accordingly on selecting India or International radio button.	rq_1		
3	Raj Travels/Book your Hotel	sc_02	Verification of check-in and check -out date list box	Verify that the check-in and check-out date should be in the valid data format(DDMONYYY). Check-out date should be greater than or equal to check in date.	rq_2		
4	Raj Travels/Book your Hotel	sc_03	Verification of rooms,adults and children list box	Verify that number of rows should coincide with number of rooms selected and the number of adults and children are within the valid data ranges.	rq_3		
5	Raj Travels/Book your Hotel	sc_04	Verification of search button	Verify that on clicking search button,the required details are displayed when data is valid and an error message is displayed when data is invalid.	rq_4		
6							

Test Cases



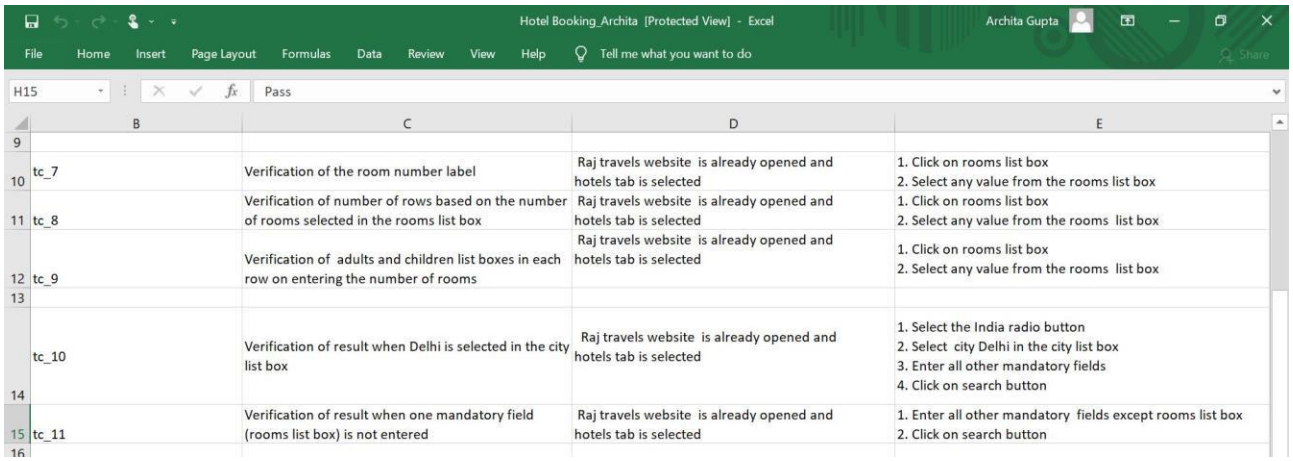
Hotel Booking_Archita [Protected View] - Excel — Archita Gupta

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

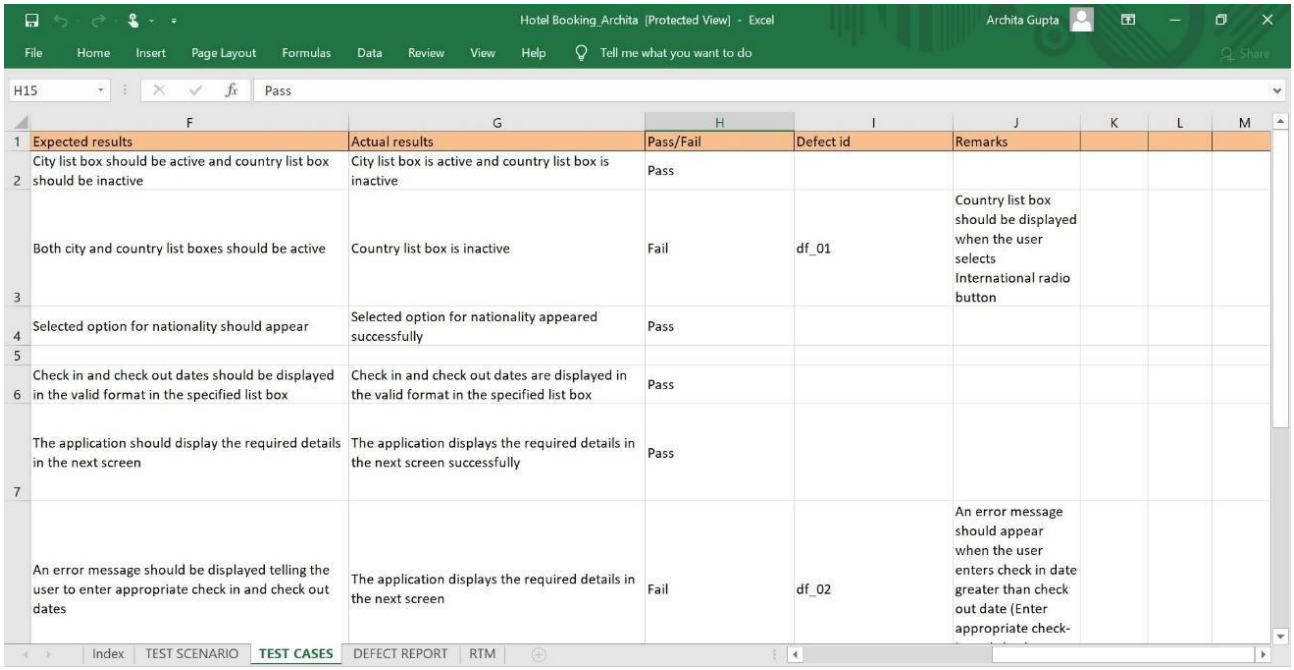
H15: Pass

	B	C	D	E
1	Test case id	Test case description	Prerequisites	Steps to execute
2	tc_1	Verification of city and country list boxes when radio button is selected as India	Raj travels website is already opened and hotels tab is selected	1. Select the India radio button
3	tc_2	Verification of city and country list boxes when radio button is selected as International	Raj travels website is already opened and hotels tab is selected	1. Select the International radio button
4	tc_3	Verification of nationality list box	Raj travels website is already opened and hotels tab is selected	1. Click on Nationality list box 2. Select any value from the drop down list
5				
6	tc_4	Verification of check in and check out dates using valid format(DDMONYYYY)	Raj travels website is already opened and hotels tab is selected	1. Enter the valid check in date in the check in list box 2. Enter the valid check out date in the check out list box
7	tc_5	Verification of results when Check in date is less than check out date	Raj travels website is already opened and hotels tab is selected	1. Enter the check in date less than check out date(check in d 10 June2021 and check out date: 20 June 2021) 2. Enter all other mandatory fields 3. Click on search button
	tc_6	Verification of results when Check in date is greater than check out date	Raj travels website is already opened and hotels tab is selected	1. Enter the check in date greater than check out date(check date: 30 June2021 and check out date: 15 June 2021) 2. Enter all other mandatory fields 3. Click on search button

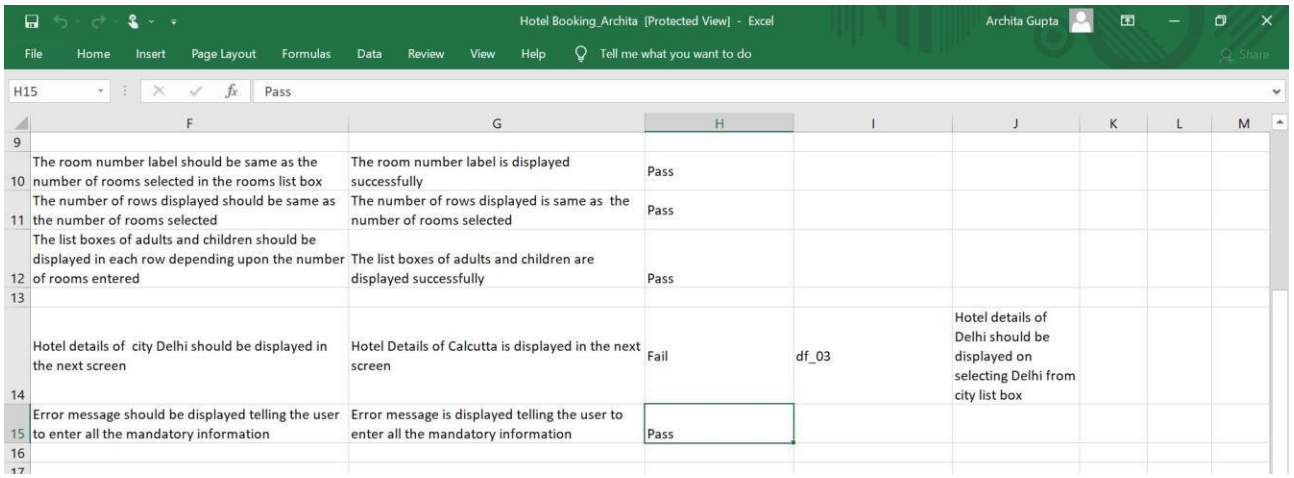
Index | TEST SCENARIO | TEST CASES | DEFECT REPORT | RTM



	B	C	D	E
9				
10	tc_7	Verification of the room number label	Raj travels website is already opened and hotels tab is selected	1. Click on rooms list box 2. Select any value from the rooms list box
11	tc_8	Verification of number of rows based on the number of rooms selected in the rooms list box	Raj travels website is already opened and hotels tab is selected	1. Click on rooms list box 2. Select any value from the rooms list box
12	tc_9	Verification of adults and children list boxes in each row on entering the number of rooms	Raj travels website is already opened and hotels tab is selected	1. Click on rooms list box 2. Select any value from the rooms list box
13				
14	tc_10	Verification of result when Delhi is selected in the city list box	Raj travels website is already opened and hotels tab is selected	1. Select the India radio button 2. Select city Delhi in the city list box 3. Enter all other mandatory fields 4. Click on search button
15	tc_11	Verification of result when one mandatory field (rooms list box) is not entered	Raj travels website is already opened and hotels tab is selected	1. Enter all other mandatory fields except rooms list box 2. Click on search button

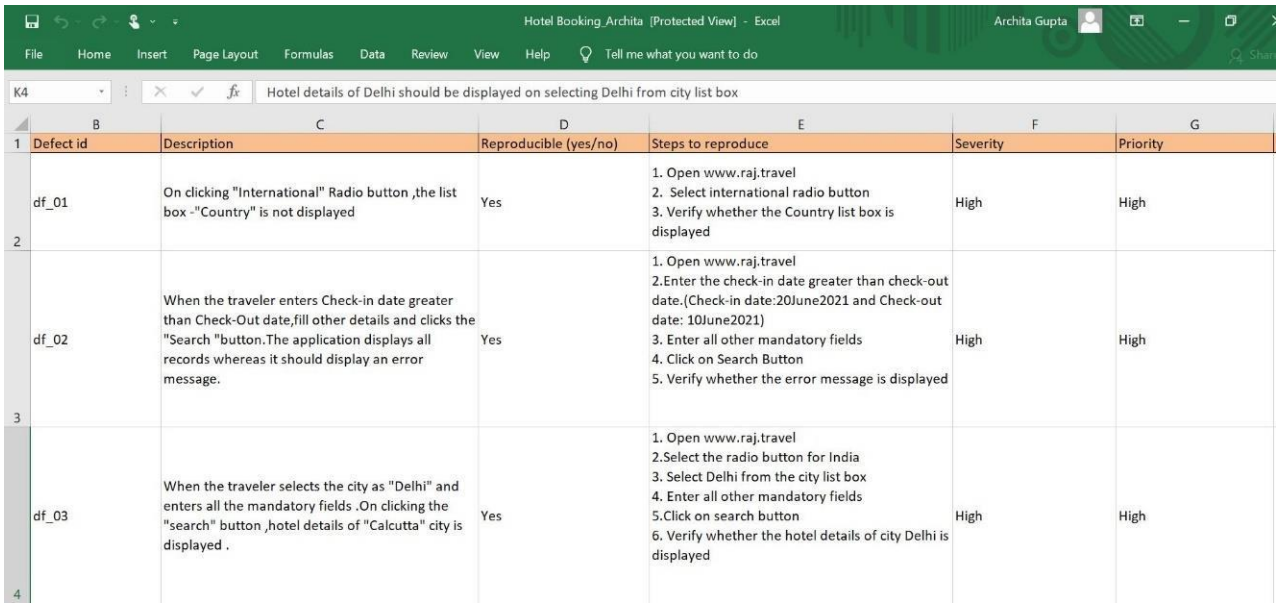


	F	G	H	I	J	K	L	M
1	Expected results	Actual results	Pass/Fail	Defect id	Remarks			
2	City list box should be active and country list box should be inactive	City list box is active and country list box is inactive	Pass					
3	Both city and country list boxes should be active	Country list box is inactive	Fail	df_01	Country list box should be displayed when the user selects International radio button			
4	Selected option for nationality should appear	Selected option for nationality appeared successfully	Pass					
5								
6	Check in and check out dates should be displayed in the valid format in the specified list box	Check in and check out dates are displayed in the valid format in the specified list box	Pass					
7	The application should display the required details in the next screen	The application displays the required details in the next screen successfully	Pass					
	An error message should be displayed telling the user to enter appropriate check in and check out dates	The application displays the required details in the next screen	Fail	df_02	An error message should appear when the user enters check in date greater than check out date (Enter appropriate check-			



	F	G	H	I	J	K	L	M
9								
10	The room number label should be same as the number of rooms selected in the rooms list box	The room number label is displayed successfully	Pass					
11	The number of rows displayed should be same as the number of rooms selected	The number of rows displayed is same as the number of rooms selected	Pass					
12	The list boxes of adults and children should be displayed in each row depending upon the number of rooms entered	The list boxes of adults and children are displayed successfully	Pass					
13								
14	Hotel details of city Delhi should be displayed in the next screen	Hotel Details of Calcutta is displayed in the next screen	Fail	df_03	Hotel details of Delhi should be displayed on selecting Delhi from city list box			
15	Error message should be displayed telling the user to enter all the mandatory information	Error message is displayed telling the user to enter all the mandatory information	Pass					
16								

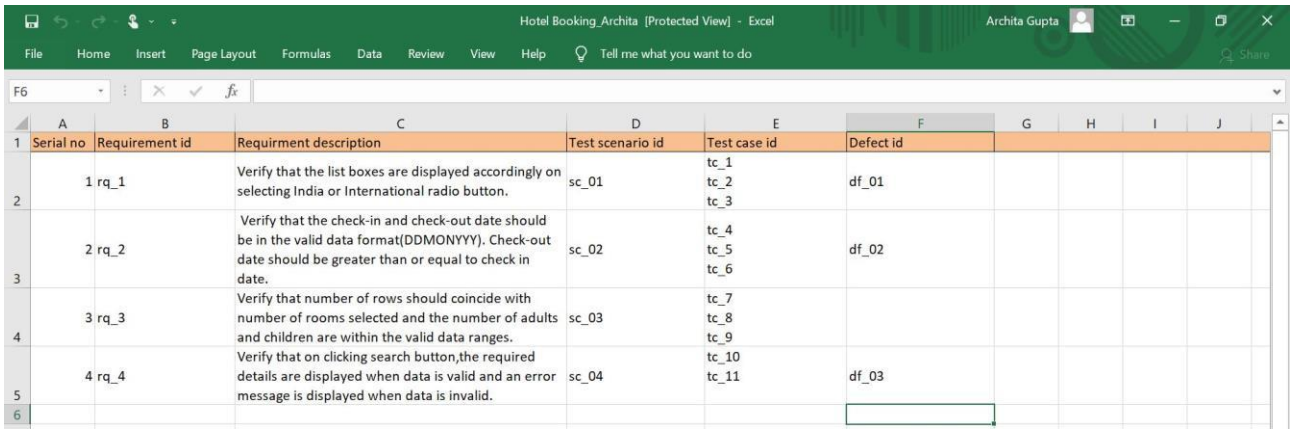
Defect Report



Defect id	Description	Reproducible (yes/no)	Steps to reproduce	Severity	Priority
df_01	On clicking "International" Radio button ,the list box -"Country" is not displayed	Yes	1. Open www.raj.travel 2. Select international radio button 3. Verify whether the Country list box is displayed	High	High
df_02	When the traveler enters Check-in date greater than Check-Out date,fill other details and clicks the "Search "button.The application displays all records whereas it should display an error message.	Yes	1. Open www.raj.travel 2.Enter the check-in date greater than check-out date.(Check-in date:20June2021 and Check-out date: 10June2021) 3. Enter all other mandatory fields 4. Click on Search Button 5. Verify whether the error message is displayed	High	High
df_03	When the traveler selects the city as "Delhi" and enters all the mandatory fields .On clicking the "search" button ,hotel details of "Calcutta" city is displayed .	Yes	1. Open www.raj.travel 2.Select the radio button for India 3. Select Delhi from the city list box 4. Enter all other mandatory fields 5.Click on search button 6. Verify whether the hotel details of city Delhi is displayed	High	High

Reported by	Reported date	Status	Remarks
Archita	24-03-2021	NEW	Country list box should be displayed when the user selects International radio button
Archita	24-03-2021	NEW	An error message should appear when the user enters check in date greater than check out date (Enter appropriate check-in and check-out dates)
Archita	24-03-2021	NEW	Hotel details of Delhi should be displayed on selecting Delhi from city list box

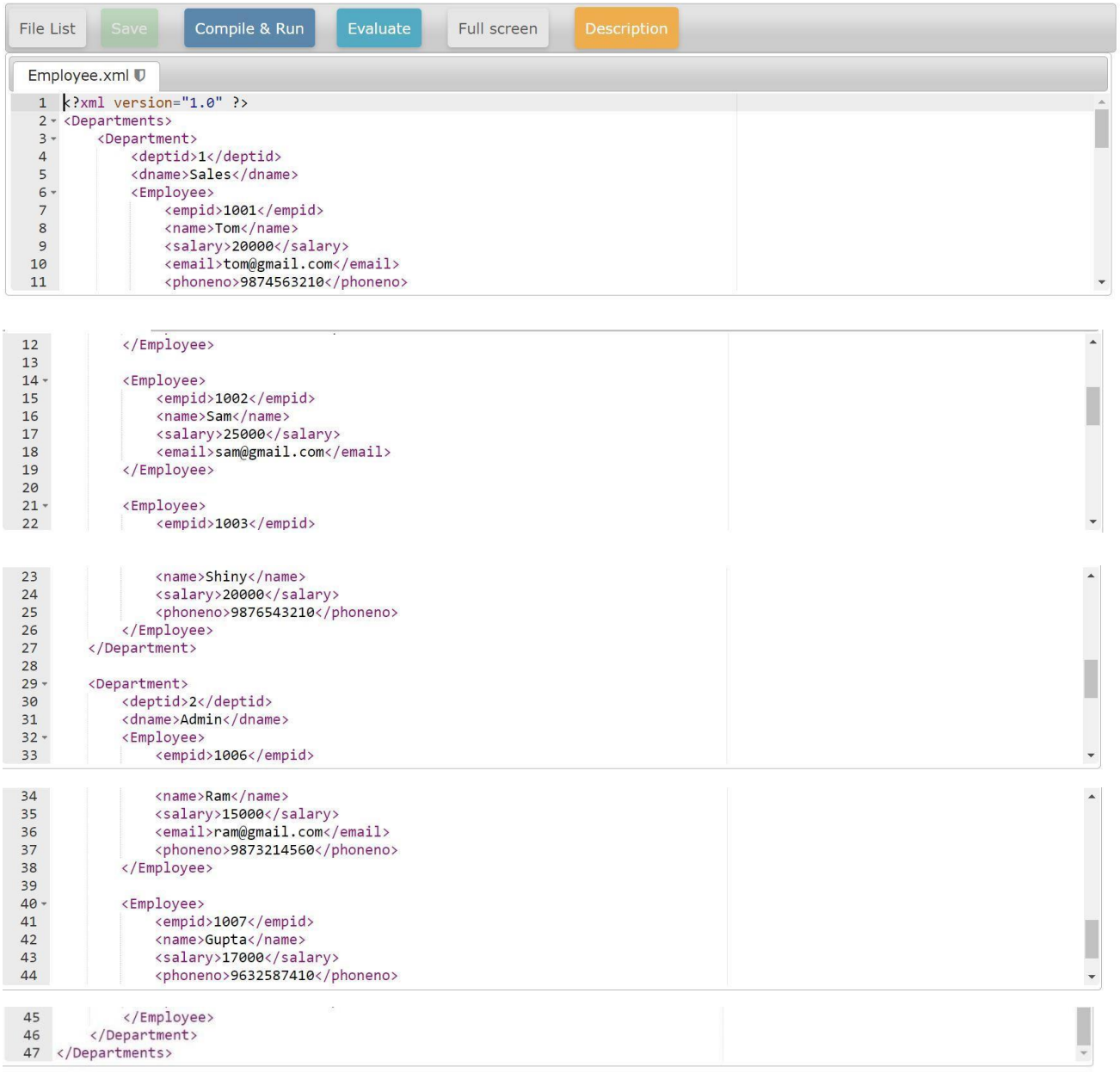
RTM (Requirement Traceability Matrix)



Serial no	Requirement id	Requirment description	Test scenario id	Test case id	Defect id
1	rq_1	Verify that the list boxes are displayed accordingly on selecting India or International radio button.	sc_01	tc_1 tc_2 tc_3	df_01
2	rq_2	Verify that the check-in and check-out date should be in the valid data format(DDMONYYY). Check-out date should be greater than or equal to check in date.	sc_02	tc_4 tc_5 tc_6	df_02
3	rq_3	Verify that number of rows should coincide with number of rooms selected and the number of adults and children are within the valid data ranges.	sc_03	tc_7 tc_8 tc_9	
4	rq_4	Verify that on clicking search button,the required details are displayed when data is valid and an error message is displayed when data is invalid.	sc_04	tc_10 tc_11	df_03

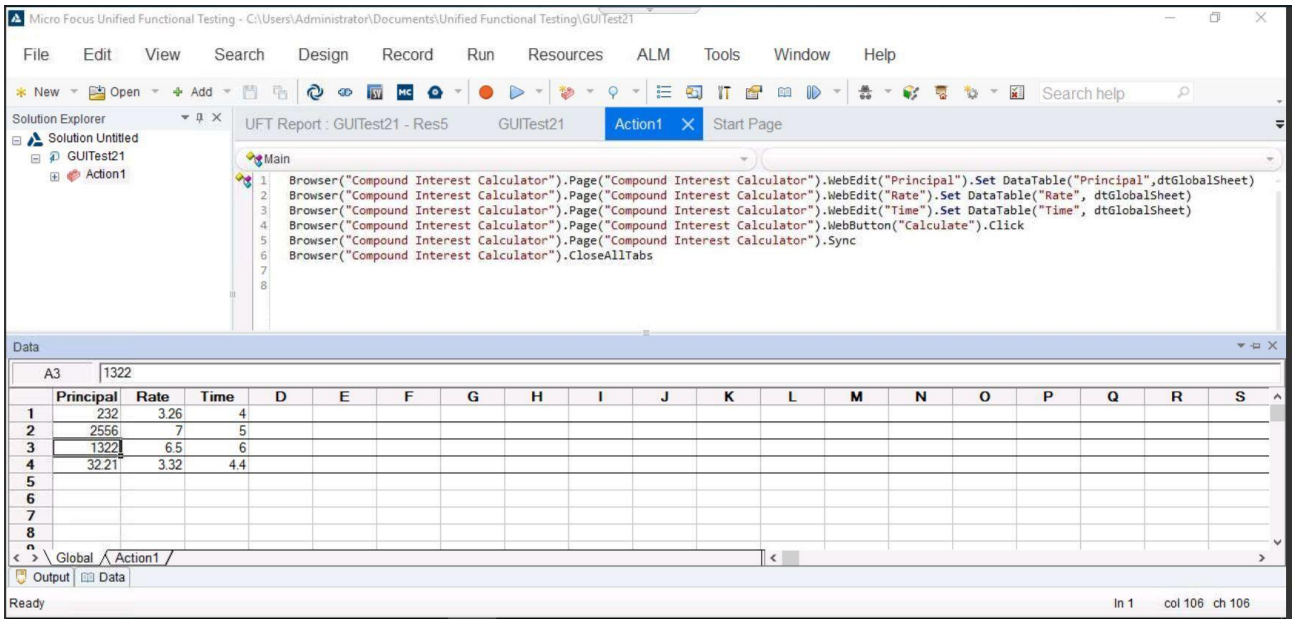
5.2 Creating a well-formed document for the given scenario using XML

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

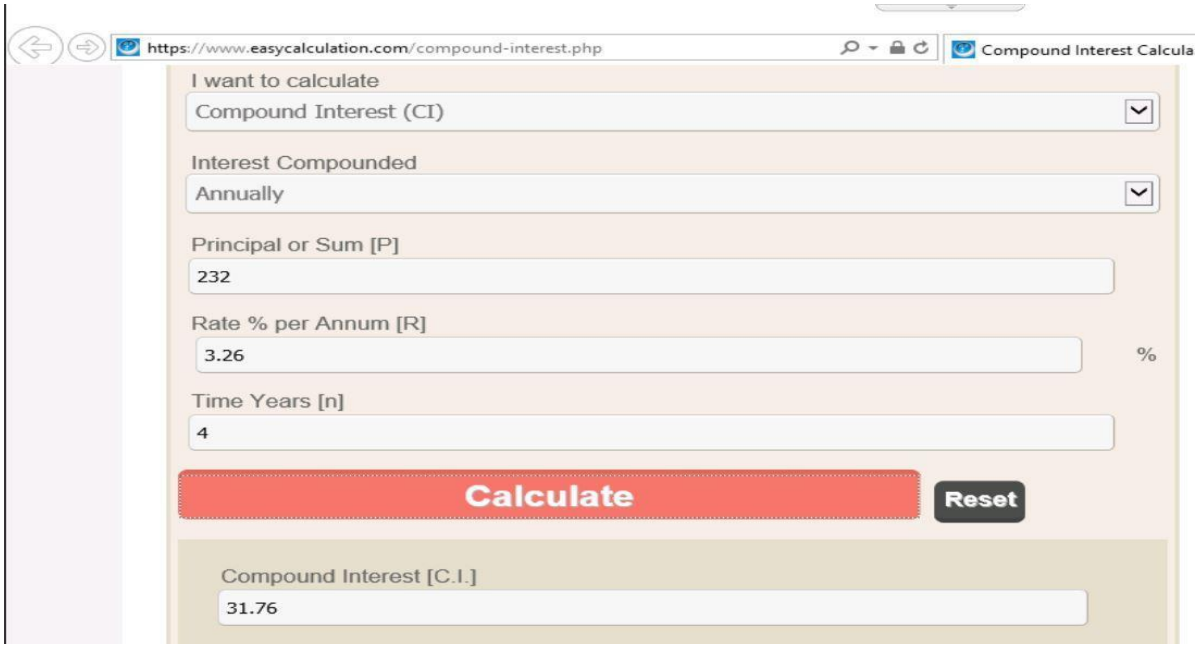


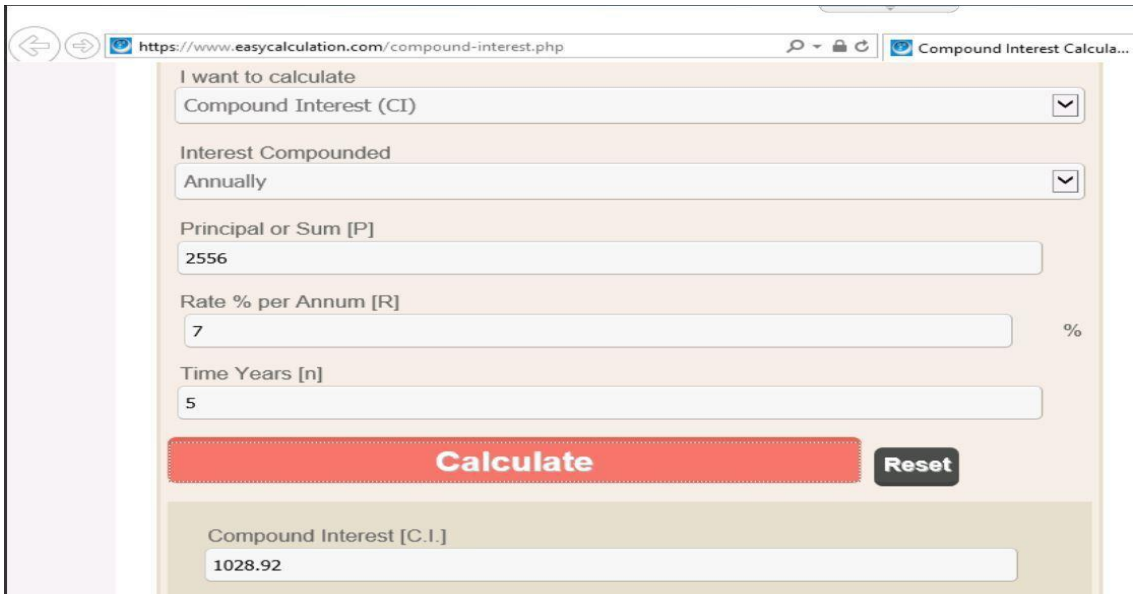
5.3 To check the functionality of calculate button for website http://easycalculation.com/compound-interest.php for different inputs using UFT Automation.

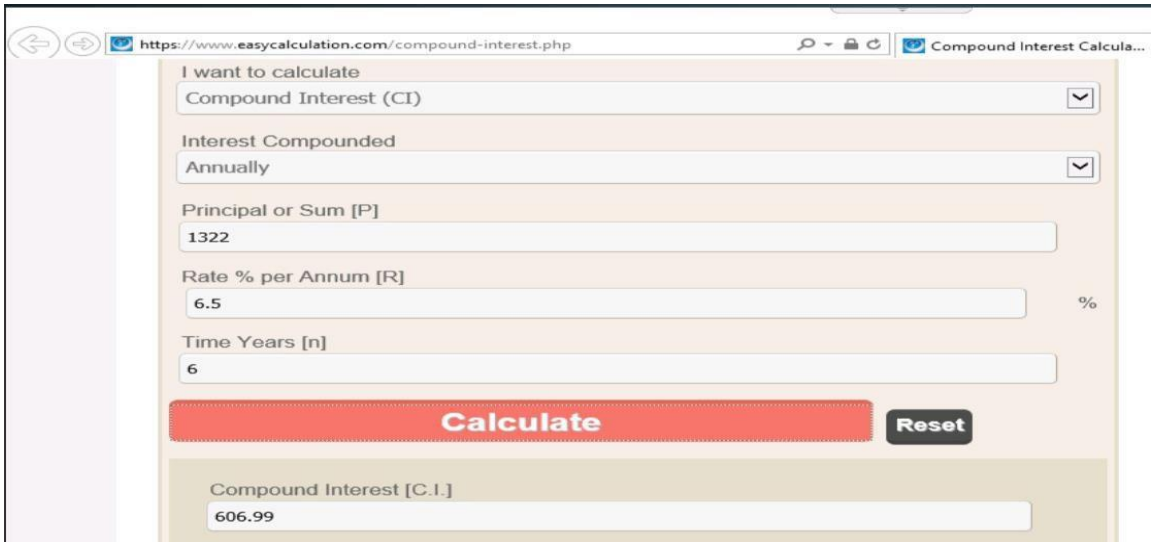
Script

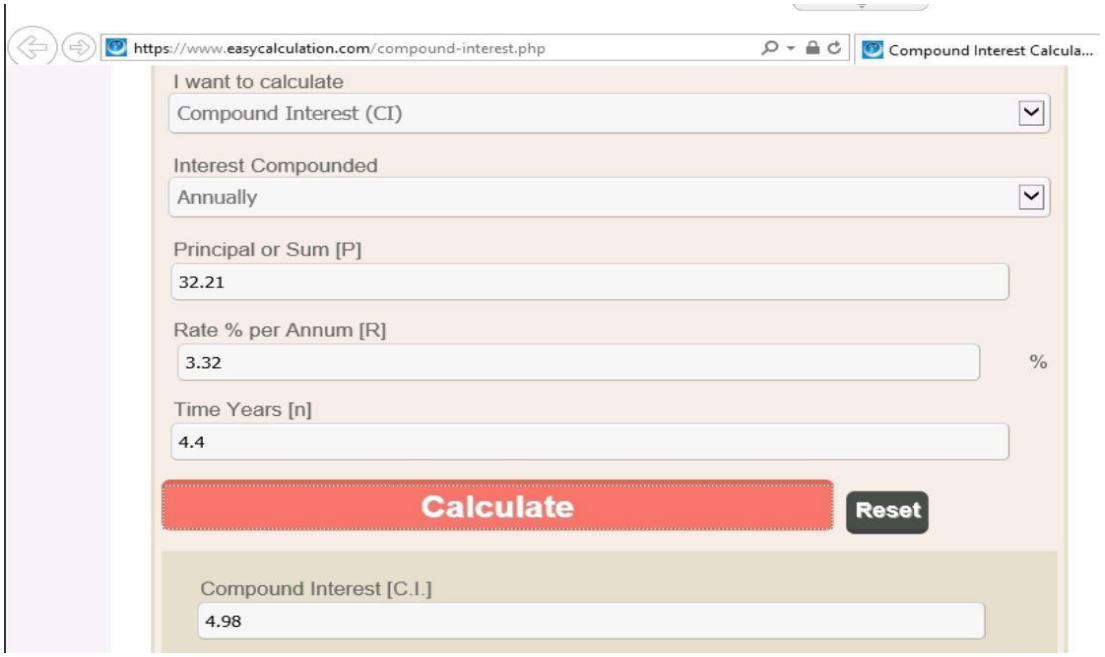


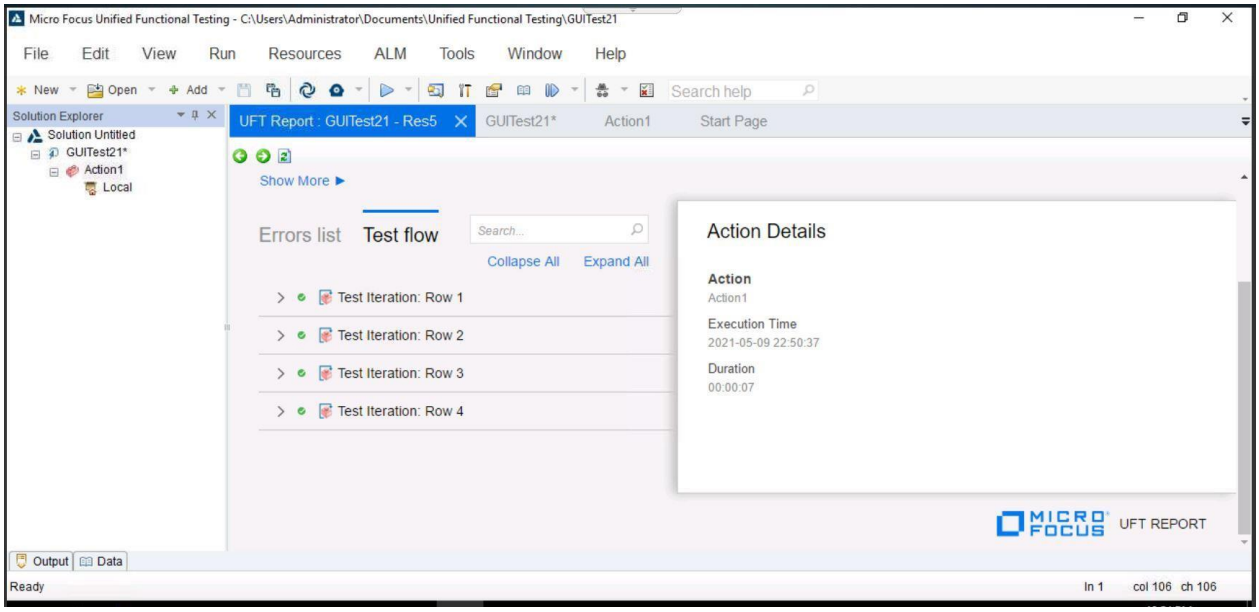
Outputs

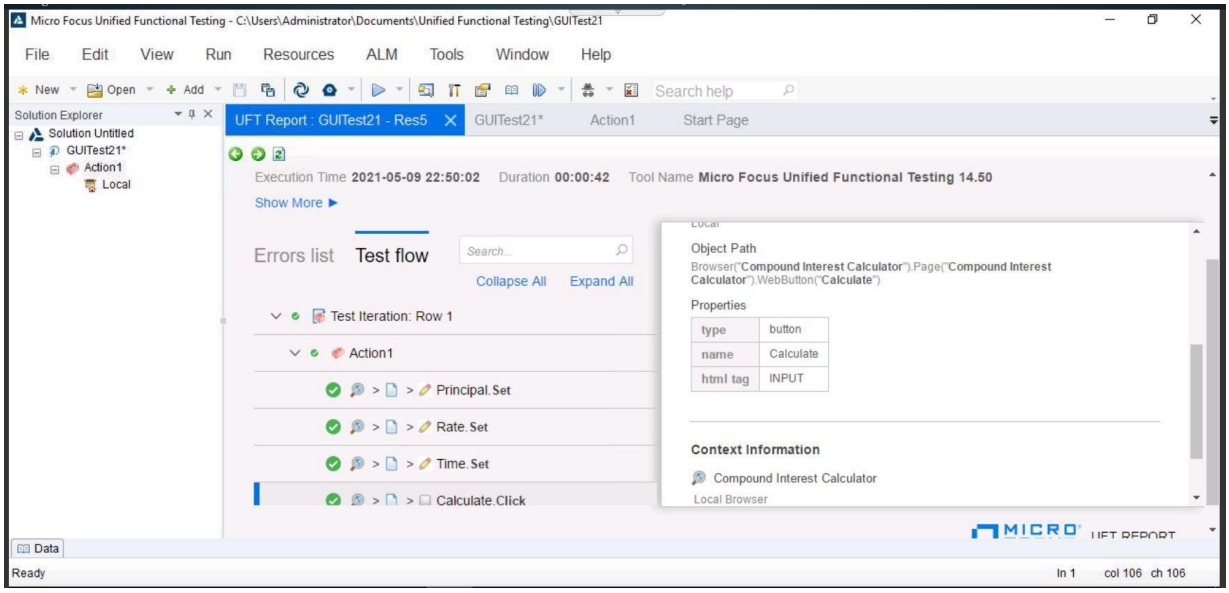



https://www.easycalculation.com/compound-interest.php
Compound Interest Calcula...
I want to calculate
Compound Interest (CI)
Interest Compounded
Annually
Principal or Sum [P]
2556
Rate % per Annum [R]
7
%
Time Years [n]
5
Calculate
Reset
Compound Interest [C.I.]
1028.92


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

Micro Focus' UFT One is tool that employs automate tests in order to find problems of a test application. Unified Functional Testing (UFT) is an acronym for "Unified Functional Testing." It was previously known as QTP (Quick Test Professional).

Functional, regression, and service testing are the most common uses for UFT. We may use UFT to automate user activities on a website or server computer program, and then test and detect defects for multiple users, various data sets, different Windows operating systems, and/or different devices using the same operations. When compared to manual testing, automation via UFT may save a lot of time and money if well designed and done. Today, UFT is one of the most commonly utilized business automated testing solutions available. It's well-known for its simplicity of use and vendor support, as well as a big community of automation experts. As a result, qualified UFT specialists have always been in high demand

Thus, using software testing techniques test cases were written for the particular application. Moreover, with the help of UFT automation tool we’re able to test the functionality of the website successfully.

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.