**Test Scenario**

A Test Scenario is any functionality that can be tested. It is also called **Test Condition or Test Possibility**. As a tester, you may put yourself in the end user’s shoes and figure out the real-world scenarios and use cases of the Application Under Test.

**What is Scenario Testing?**

Scenario Testing is a variant of Software Testing where Scenarios are Used for Testing. Scenarios help in an Easier Way of Testing of the more complicated Systems

Let’s study this with the help of the video below -

**Why create Test Scenarios?**

Test Scenarios are created for following reasons,

* Creating Test Scenarios ensures complete Test Coverage
* Test Scenarios can be approved by various stakeholders like Business Analyst, Developers, Customers to ensure the Application Under Test is thoroughly tested. It ensures that the software is working for the most common use cases.
* They serve as a quick tool to determine the testing work effort and accordingly create a proposal for the client or organize the workforce.
* They help determine the most important end-to-end transactions or the real use of the software applications.
* For studying the end-to-end functioning of the program, Test Scenario is critical.

**When not create Test Scenario?**

Test Scenarios may not be created when

* The Application Under Test is complicated, unstable and there is a time crunch in the project.
* Projects that follow Agile Methodology like Scrum, Kanban may not create Test Scenarios.
* Test Scenario may not be created for a new bug fix or regression testing. In such cases, Test Scenarios must be already heavily documented in the previous test cycles. This is especially true for Maintenance projects.

**How to create a Test Scenario**

As a tester, you can follow these five steps to create Test Scenarios-

* **Step 1**: Read the Requirement Documents like BRS, SRS, FRS, of the System Under Test (SUT).  You could also refer uses cases, books, manual, etc. of the application to be tested.
* **Step 2**: For each requirement, figure out possible users actions and objectives. Determine the technical aspects of the requirement. Ascertain possible scenarios of system abuse and evaluate users with hacker's mindset.
* **Step 3:** After reading the Requirements Document and doing your due Analysis, list out different test scenarios that verify each feature of the software.
* **Step 4:** Once you have listed all possible Test Scenarios, a Traceability Matrix is created to verify that each & every requirement has a corresponding Test Scenario
* **Step 5:**The scenarios created are reviewed by your supervisor. Later, they are also reviewed by other Stakeholders in the project.

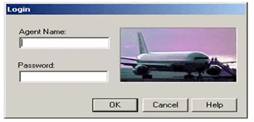
**Tips to Create Test Scenarios**

* Each Test Scenario should be tied to a minimum of one Requirement or User Story as per the Project Methodology.
* Before creating a Test Scenario that verifies multiple Requirements at once, ensure you have a test scenario that checks that requirement in isolation.
* Avoid creating overly complicated Test Scenarios spanning multiple Requirements.
* The number of scenarios may be large, and it is expensive to run them all. Based on customer priorities only run selected Test Scenarios

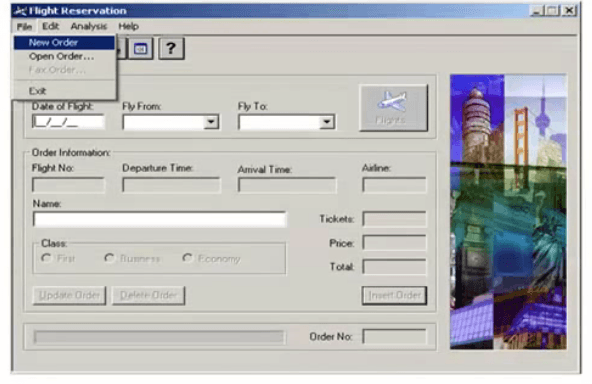
**Example 1: Test Scenario for Flight Reservation**

For the Flight Reservation Application, a few test scenarios would be

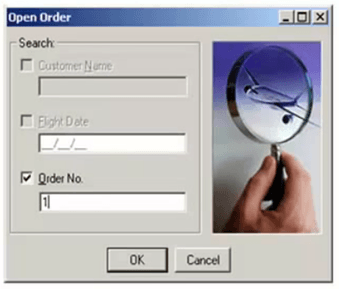
**Test Scenario 1:**Check the Login Functionality



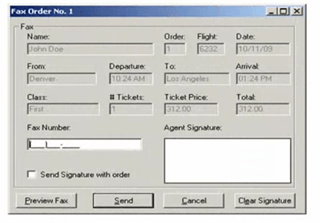
**Test Scenario 2:**Check that a New Order can be created



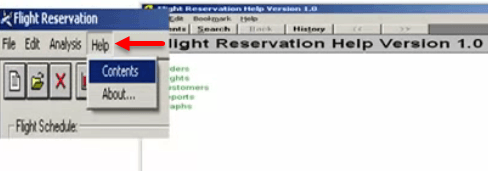
**Test Scenario 3:**Check that an existing Order can be opened



**Test Scenario 4:**Check that a user, can FAX an order



**Test Scenario 5:**Check that the information displayed in the HELP section is correct



**Test Scenario 6:**Check that the information displayed in About section, like version, programmer name, copy right information is correct



Apart from these six scenarios here is the list of all other scenarios

* Update Order
* Delete Order
* Check Reports
* Check Graphs and so on.

Next, we have already learned **exhaustive testing is not possible**. Suppose you have time only to execute 4 out of these 6 scenarios which two low priority scenarios of these six will you eliminate. Think, your time starts now

I am sure most of you would have guessed scenarios 5 & 6 since they are not the core functionality of the application. This is nothing but **Test Prioritization**.

**Example 2:  Test Scenarios for a Banking Site**

**Test Scenario 1**: Check the Login and Authentication Functionality

**Test Scenario 2**: Check Money Transfer can be done

**Test Scenario 3**: Check Account Statement can be viewed

**Test Scenario 4**: Check Fixed Deposit/Recurring Deposit can be created

And so on…