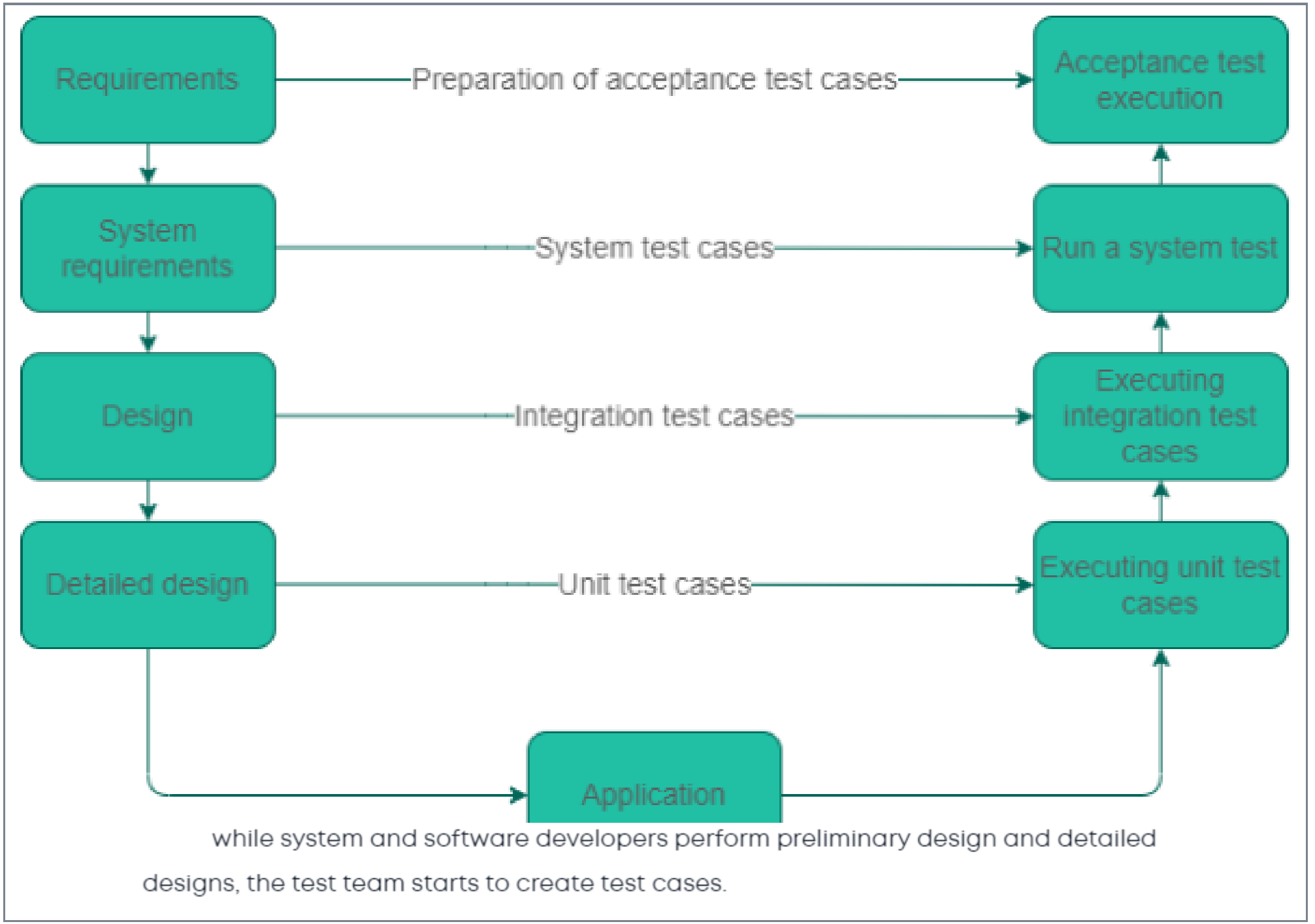
Test Cases – Test Scenario

Full proof document

What is a Test Case?

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Customer Login with valid Data	1. Go to site `http://demo.guru99.com` 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = pass99	User should Login into an application	As Expected	Pass
TU02	Check Customer Login with invalid Data	1. Go to site `http://demo.guru99.com` 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = glass99	User should not Login into an application	As Expected	Pass



while system and software developers perform preliminary design and detailed designs, the test team starts to create test cases.

Test Case

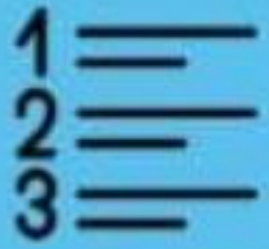
A set of preconditions, inputs, actions (where applicable), expected results and post conditions, developed based on test conditions.

Test Script

A sequence of instructions for the execution of a test.

Test Suite

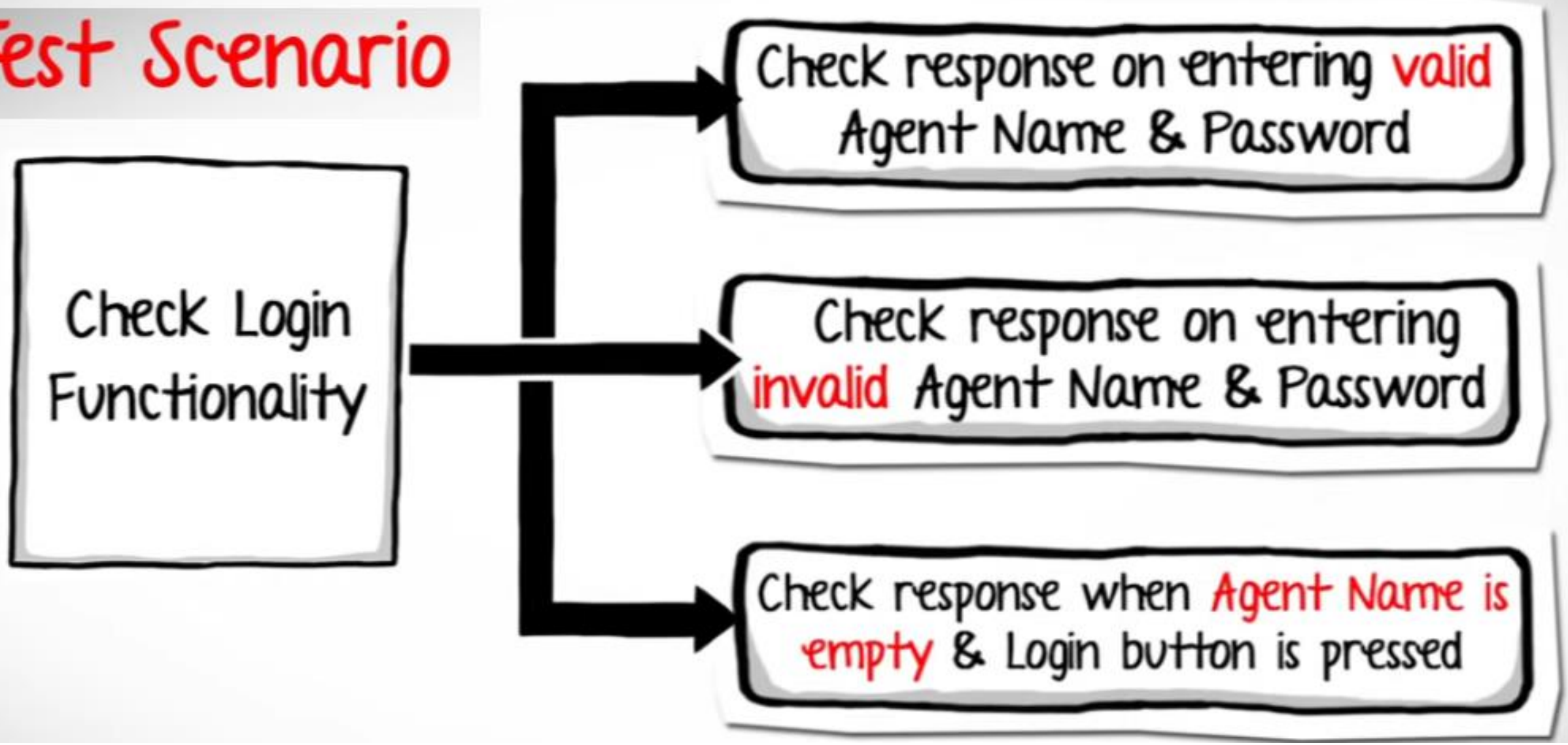
groups of test scripts, as well as a test execution schedule.



Test Charter

An instruction of test goals and possible test ideas on how to test. Documentation of test activities in session-based exploratory testing.

Test Scenario



Test Case Content

A test case should include

1. the purpose and conditions for its execution,
2. a step-by-step setup of the test environment,
3. input data,
4. the expected result,
5. the actual result,
6. a version description of the software,
7. the software's working environment, and
8. the test ID.

Test Scenario	Test Case	Test Data	Expected Result
Check Login Functionality	Check response on Entering valid Agent Name & Password	Agent Name : guru99 Password : MERCURY Agent Name : guru Password : MERcury Agent Name : 9999 Password : mercury	Login must be Successful

Test Scenario	Test Case	Test Steps	Test Data	Expected Result
Check Login Functionality	Check response on Entering valid Agent Name & Password	1. Launch Application 2. Enter Agent Name 3. Enter Password 4. Click OK button	Agent Name : guru99 Password : MERCURY	Login must be successful.

Test Scenario	Test Case	Pre conditions	Test Step	Test Data	Expected Result
Check Login Functionality	Check response on Entering valid Agent Name & Password	Flight Reservation Application must be installed	1. Launch Application 2. Enter Agent Name 3. Enter Password 4. Click OK button	Agent Name : guru99 Password : MERCURY	Login must be successful.

For a Test Scenario: Check Login Functionality there many possible test cases are:

- Test Case 1: Check results on entering valid User Id & Password
- Test Case 2: Check results on entering Invalid User ID & Password
- Test Case 3: Check response when a User ID is Empty & Login Button is pressed, and many more

Step 1) A simple test case to explain the scenario would be

Test Case #	Test Case Description
1	Check response when valid email and password is entered

Step 2) Test the Data.
In order to execute the test case, you would need Test Data. Adding it below

Test Case #	Test Case Description	Test Data
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h

Step 3) Perform actions.

In order to execute a test case, a tester needs to perform a specific set of actions on the AUT. This is documented as below:

Test Case #	Test Case Description	Test Steps	Test Data
1	Check response when valid email and password is entered	1) Enter Email Address 2) Enter Password 3) Click Sign in	Email: guru99@email.com Password: lNf9^Oti7^2h

Step 4) Check behavior of the AUT.

The goal of test cases in software testing is to check behavior of the AUT for an expected result. This needs to be documented as below

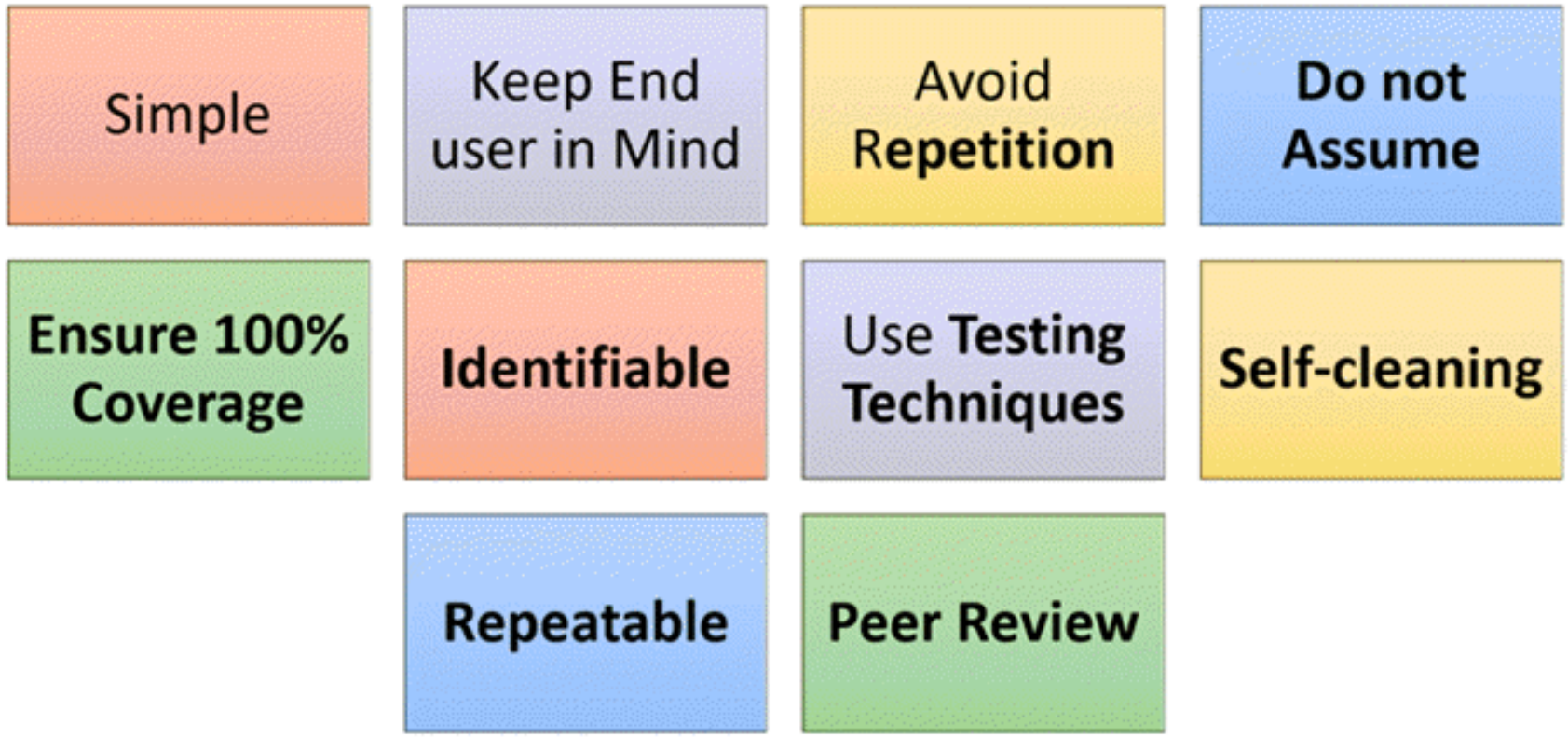
Test Case #	Test Case Description	Test Data	Expected Result
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h	Login should be successful

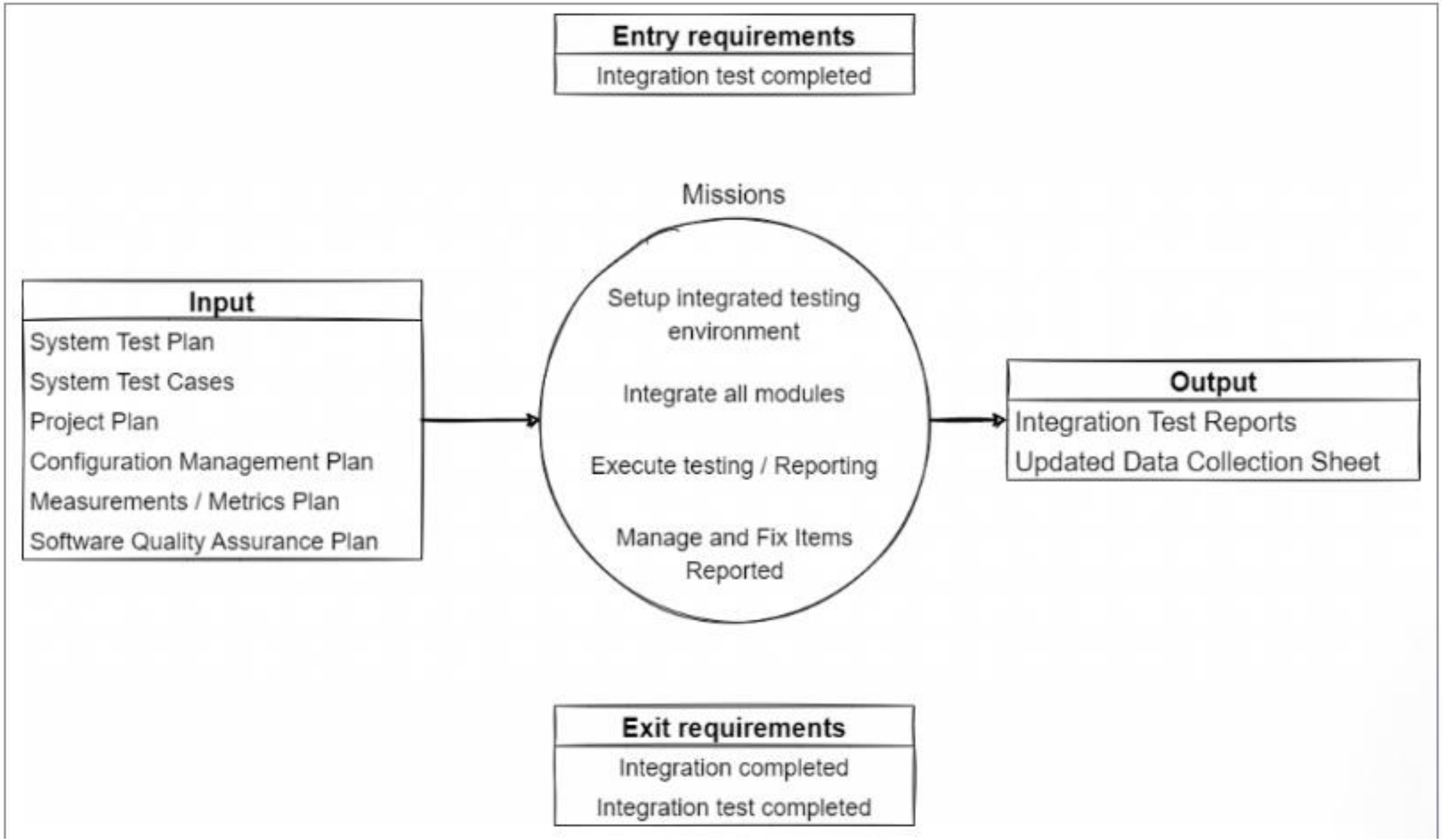
Test Case #	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid email and password is entered	Email: guru99@email.com Password: lNf9^Oti7^2h	Login should be successful	Login was successful	Pass

Step 5) That apart your test case -may have a field like,

Pre – Condition which specifies things that must be in place before the test can run. For our test case, a pre-condition would be to have a browser installed to have access to the site under test. A test case may also include Post – Conditions which specifies anything that applies after the test case completes. For our test case, a postcondition would be time & date of login is stored in the database

Best Practice for writing good Test Case.




Entry requirements
Integration test completed
Missions
Input
System Test Plan
System Test Cases
Project Plan
Configuration Management Plan
Measurements / Metrics Plan
Software Quality Assurance Plan
Setup integrated testing environment
Integrate all modules
Execute testing / Reporting
Manage and Fix Items Reported
Output
Integration Test Reports
Updated Data Collection Sheet
Exit requirements
Integration completed
Integration test completed

	Test Cases	Test Scenarios
What it is =>	A concept which provides detailed information what to test, steps to be taken and expected result of the same	A concept which provides one-line information about what to test.
It's about =>	It's more about documenting details.	It's more about thinking and discussing details.
Importance =>	It's important when testing is off shored and development is onsite. Writing test cases with details will help both dev and QA team in sync.	It's important when time is less and most of the team members can agree / understand the details from one-liner scenario.
Advantage =>	One time documentation of all the test cases is beneficial to track 1000s rounds of regression testing in future. Most of the time, its helpful while bug reporting. Tester just need to give reference of test case ID and does not require mentioning each and every minute detail.	A time saver and idea generation activity, preferred by new generation software testing community. Modification and addition is simple and not specific to a person. For a huge project, where group of people know specific modules only, this activity gives a chance to everyone to look into other modules and brain storm and

Beneficial to =>	A full-proof test case document is a life line for new tester.	Good test coverage can be achieved by dividing application in test scenarios and it reduces repeatability and complexity of product
Disadvantage =>	Time and money consuming as it requires more resources to detail out everything about what to test and how to test	If created by specific person, the reviewer or the other user might not sync the exact idea behind it. Need more discussions and team efforts.

Test Case Design Environment

1. **Test data generators—**This software produces the test input data necessary to execute test cases.
2. **Debugger—**This software enables errors encountered during the tests to be found on the source code.
3. **Emulators and simulators—**These software types imitate the real hardware that the software needs; they provide the hardware data necessary for the tests.
4. **Stubs and drivers—**These small pieces of software replace nonfunctional system components required for tests.

Test Case Management Tools

Test management tools are the automation tools that help to manage and maintain the Test Cases. Main Features of a test case management tool are

1. **For documenting Test Cases:** With tools, you can expedite Test Case creation with use of templates
2. **Execute the Test Case and Record the results:** Test Case can be executed through the tools and results obtained can be easily recorded.
3. **Automate the Defect Tracking:** Failed tests are automatically linked to the bug tracker, which in turn can be assigned to the developers and can be tracked by email notifications.
4. **Traceability:** Requirements, Test cases, Execution of Test cases are all interlinked through the tools, and each case can be traced to each other to check test coverage.
5. **Protecting Test Cases:** Test cases should be reusable and should be protected from being lost or corrupted due to poor version control. Test Case Management Tools offer features like

- Naming and numbering conventions
- Versioning
- Read-only storage
- Controlled access
- Off-site backup

A basic example of test case design

Let us take an example of any e-commerce app or website(like Amazon or Flipkart)for test case design. We want to ensure users can quickly checkout and make payments without issues. Here we test for 1 product in the cart; we will see later that this test case design technique is a boundary value analysis technique.

Title: Test that user can complete the checkout process when there is 1 item in the card.

Description: Ensure users can checkout and make payments without issues on the website/app

Preconditions: The user is already logged in

Assumptions: They are using a supported device or browser to log in.

Test Steps:

1. Open the app/website.
2. Go to 1 product
3. Add the product to the cart.
4. Check out the item in the cart.
5. Add address information for delivery
6. Add payment information
7. Complete the checkout process.

Expected Result: The checkout process should be complete, and the user should receive confirmation.