**1 Equivalence and Boundary Value**

**Example 1: Equivalence and Boundary Value**

* Let’s consider the behavior of Order Pizza Text Box Below
* Pizza values 1 to 10 is considered valid. A success message is shown.
* While value 11 to 99 are considered invalid for order and an error message will appear, **“Only 10 Pizza can be ordered”**

**Order Pizza:**

**Here is the test condition**

1. Any Number greater than 10 entered in the Order Pizza field(let say 11) is considered invalid.
2. Any Number less than 1 that is 0 or below, then it is considered invalid.
3. Numbers 1 to 10 are considered valid
4. Any 3 Digit Number say -100 is invalid.

We cannot test all the possible values because if done, the number of test cases will be more than 100. To address this problem, we use equivalence partitioning hypothesis where we divide the possible values of tickets into groups or sets as shown below where the system behavior can be considered the same.

A white sheet with text

Description automatically generated with medium confidence

The divided sets are called Equivalence Partitions or Equivalence Classes. Then we pick only one value from each partition for testing. The hypothesis behind this technique is **that if one condition/value in a partition passes all others will also pass**. Likewise**, if one condition in a partition fails, all other conditions in that partition will fail**.

A diagram of a number of partitions

Description automatically generated

**2 Boundary Value Analysis**– in Boundary Value Analysis, you test boundaries between equivalence partitions

A screenshot of a computer

Description automatically generated

In our earlier equivalence partitioning example, instead of checking one value for each partition, you will check the values at the partitions like 0, 1, 10, 11 and so on. As you may observe, you test values at**both valid and invalid boundaries**. Boundary Value Analysis is also called**range checking**.

Equivalence partitioning and boundary value analysis(BVA) are closely related and can be used together at all levels of testing.

**Example 2: Equivalence and Boundary Value**

Following password field accepts minimum 6 characters and maximum 10 characters

That means results for values in partitions 0-5, 6-10, 11-14 should be equivalent

**Enter Password:**

|  |  |  |
| --- | --- | --- |
| **Test Case #** | **Test Case Description** | **Expected Outcome** |
| 1 | Enter 0 to 5 characters in password field | System should not accept |
| 2 | Enter 6 to 10 characters in password field | System should accept |
| 3 | Enter 11 to 14 character in password field | System should not accept |

**Examples 3: Input Box should accept the Number 1 to 10**

Here we will see the Boundary Value Test Cases

|  |  |
| --- | --- |
| **Test Case Description** | **Expected Outcome** |
| Boundary Value = 0 | System should NOT accept |
| Boundary Value = 1 | System should accept |
| Boundary Value = 2 | System should accept |
| Boundary Value = 9 | System should accept |
| Boundary Value = 10 | System should accept |
| Boundary Value = 11 | System should NOT accept |

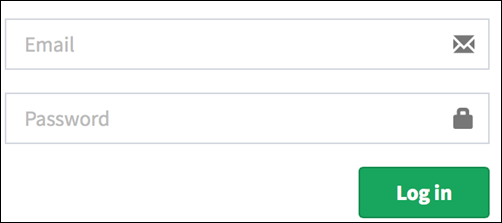
**Why Equivalence & Boundary Analysis Testing**

1. This testing is used to reduce a very large number of test cases to manageable chunks.
2. Appropriate for calculation-intensive applications with a large number of variables/inputs

# 3 Decision Table Testing (Example)

### Example 1: How to make Decision Base Table for Login Screen

Let’s create a decision table for a login screen.

[](https://www.guru99.com/images/1/120817_0759_DecisionTab1.png)

The condition is simple if the user provides the correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Conditions** | **Rule 1** | **Rule 2** | **Rule 3** | **Rule 4** |
| **Username (T/F)** | F | T | F | T |
| **Password (T/F)** | F | F | T | T |
| **Output (E/H)** | E | E | E | H |

**Legend:**

* **T** – Correct username/password
* **F** – Wrong username/password
* **E** – Error message is displayed
* **H** – Home screen is displayed

**Interpretation:**

* **Case 1** – Username and password both were wrong. The user is shown an error message.
* **Case 2** – Username was correct, but the password was wrong. The user is shown an error message.
* **Case 3** – Username was wrong, but the password was correct. The user is shown an error message.
* **Case 4** – Username and password both were correct, and the user navigated to the homepage

## Advantages of Decision Table Testing

* When the system behavior is different for different inputs and not the same for a range of inputs, both equivalent partitioning, and boundary value analysis won’t help, but a decision table can be used.
* The representation is simple so that it can be easily interpreted and is used for development and business as well.
* This table will help to make effective combinations and can ensure better coverage for testing
* Any complex business conditions can be easily turned into decision tables
* In a case we are going for 100% coverage typically when the input combinations are low, this technique can ensure the coverage.

## Disadvantages of Decision Table Testing

The main disadvantage is that when the number of inputs increases the table will become more complex

4 State Transition **Example:**

In the following example, if the user enters a valid password in any of the first three attempts the user will be able to log in successfully. If the user enters the invalid password in the first or second try, the user will be prompted to re-enter the password. When the user enters password incorrectly 3rd time, the action has taken, and the account will be blocked.

## State Transition Diagram

A diagram of a computer flowchart

Description automatically generated

In this diagram when the user gives the correct PIN number, he or she is moved to Access granted state. Following Table is created based on the diagram above-

## State Transition Table

|  |  |  |
| --- | --- | --- |
| States | **Correct PIN** | **Incorrect PIN** |
| **S1) Start** | **S2** | **S2** |
| **S2) 1st attempt** | **S5** | **S3** |
| **S3) 2nd attempt** | **S5** | **S4** |
| **S4) 3rd attempt** | **S5** | **S6** |
| **S5) Access Granted** | **–** | **–** |
| **S6) Account blocked** | **–** | **–** |

In the above-given table when the user enters the correct PIN, the state is transitioned to Access granted. And if the user enters an incorrect password, he or she is moved to next state. If he does the same 3rd time, he will reach the account blocked state.

### 5 Use Case Testing

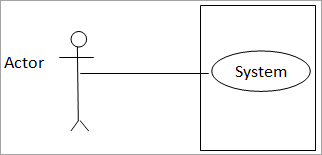
**What Is Use Case?**

Use Case represents the different ways in which the system can be used by the users. The use case divides the system behavior into scenarios, such that each scenario performs some useful action from the user’s point of view.

Each scenario may involve a single message or multiple message exchanges between the user and the system to complete itself.

**Representation of Use cases:**

A use case is represented with actors and systems. The actor represents the user. In the Use case diagram, Stick represents the actor, and ellipses in the rectangle represent the Use case.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/12/Representation-of-Use-cases.png)

**Use Case Testing:**

Use Case testing is a technique wherein the use cases identify all the test cases that cover the complete system.

**Example:**

When a new user tries to create an account on a xyz website.

| **Main Success Scenario** | **Step** | **Description** |
| --- | --- | --- |
| A: Actor S: System | 1 | A:Enters mobile no |
|  | 2 | S: Verifies mobile number & asks to put OTP |
|  | 3 | A: Receives OTP and fill it on the site. |
|  | 4 | S: Verifies OTP and asks to put password |
|  | 5 | A: Puts the password |
|  | 6 | S: Verifies the password |
|  | 7 | A:clicks on sign up to create a account |
| Extensions | 2a | Incorrect mobile number |
|  |  | A:clicks on sign up to |
|  | 2b | create a account |
|  |  | S: Displays message That user already exists |
|  | 4a | Incorrect OTP |
|  |  | S: Asks to enter valid OTP |
|  | 4b | correct OTP |
|  |  | S:accepts & asks to provide password |
|  | 6a | Password does not match the criteria |
|  |  | S: Displays message “Incorrect Password” |

First, a positive scenario that carries end-to-end testing is tested i.e. complete scenario of creating an account is tested. Afterward, negative testing starts i.e. if the mobile number is incorrect, the user will get a message for the same.

Once the user provides a correct mobile number, the system will send OTP to the user’s mobile number and the user will have to provide the same on the OTP field of the signup screen.

In case of incorrect OTP, the system asks to provide the correct OTP.

Once correct OTP is provided, the system asks to fill the password, and the password should match the criteria defined else it asks to fill the password again. Once the correct password is provided the system creates the user’s account by clicking sign up.

## 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. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

**Test Scenario Vs Test Case**

For a [Test Scenario](https://www.guru99.com/test-scenario.html): 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

**The format of Standard Test Cases**

Below is a format of a standard login Test cases example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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://abc.com 2. Enter UserId 3. Enter Password 4. Click Submit | Userid = faisal Password = bahadur | User should Login into an application | As Expected | Pass |
| TU02 | Check Customer Login with invalid Data | 1. Go to site http://abc.com 2. Enter UserId 3. Enter Password 4. Click Submit | Userid = faisal Password = bahadir | User should not Login into an application | As Expected | Pass |

This entire table may be created in Word, Excel or any other Test management tool. That’s all to Test Case Design