**State Transition Table**

* **State Transition Table** is a test design technique used to model the behaviour of a system based on its states and transitions.
* It is particularly useful in scenarios where the system's behaviour changes based on previous events, such as a **login system**.

**Example of State Transition Table for Login Test Design**

In the context of a **login system**, a user can be in different states, and based on the actions they take (e.g., entering a username and password), the system transitions between states.

**Step 1: Define the States**

Let’s define the states for a user interacting with the login system:

1. **Initial State (S1)**: The user has not attempted to log in yet.
2. **Logged in State (S2)**: The user has successfully logged in.
3. **Failed Login State (S3)**: The user has failed the login attempt, either due to an invalid username, invalid password, or other reasons.
4. **Locked Account State (S4)**: The user's account is locked (after multiple failed login attempts).

**Step 2: Define the Inputs or Events**

The inputs (or events) that trigger state transitions are:

* **Enter Username**
* **Enter Password**
* **Click Login Button**
* **Invalid Login Attempt**
* **Captcha Validation**

**Step 3: Create the State Transition Table**

Here’s how the system behaves depending on the combination of inputs (actions) and current state:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Step 4: Derive Test Cases from the State Transition Table**

From the state transition table, we can now derive the corresponding test cases that ensure all transitions and scenarios are tested.

**Example Test Cases**

1. **Test Case 1: Successful Login**
   * **Initial State (S1)** → Enter valid Username & Password → **S2: Logged In**
   * **Expected Outcome**: User successfully logs in.
2. **Test Case 2: Invalid Username**
   * **Initial State (S1)** → Enter invalid Username → **S3: Failed Login**
   * **Expected Outcome**: Error message: "Invalid username."
3. **Test Case 3: Invalid Password**
   * **Initial State (S1)** → Enter valid Username, invalid Password → **S3: Failed Login**
   * **Expected Outcome**: Error message: "Invalid password."
4. **Test Case 4: Multiple Failed Login Attempts (Locked Account)**
   * **Initial State (S1)** → Enter invalid Username & Password (Failed 3 times) → **S4: Locked Account**
   * **Expected Outcome**: Error message: "Account locked."
5. **Test Case 5: Login After Account Lock (S4)**
   * **Locked Account (S4)** → Enter valid Username & Password → **S4: Locked Account**
   * **Expected Outcome**: Error message: "Account is locked, contact support."
6. **Test Case 6: Successful Login After Failure (S3 to S2)**
   * **Failed Login (S3)** → Enter valid Username & Password → **S2: Logged In**
   * **Expected Outcome**: User logs in successfully.

**Conclusion**

By using the **State Transition Table**, we can clearly define the different states of the system, the inputs that cause state transitions, and the expected outcomes. This approach ensures comprehensive testing of the login functionality, covering valid and invalid login attempts, account locking, and proper transitions between states.