2.3 Decision Table Testing

Decision Table method is a black-box testing technique that visually represents combinations of input conditions and their corresponding expected outputs.

How it works:

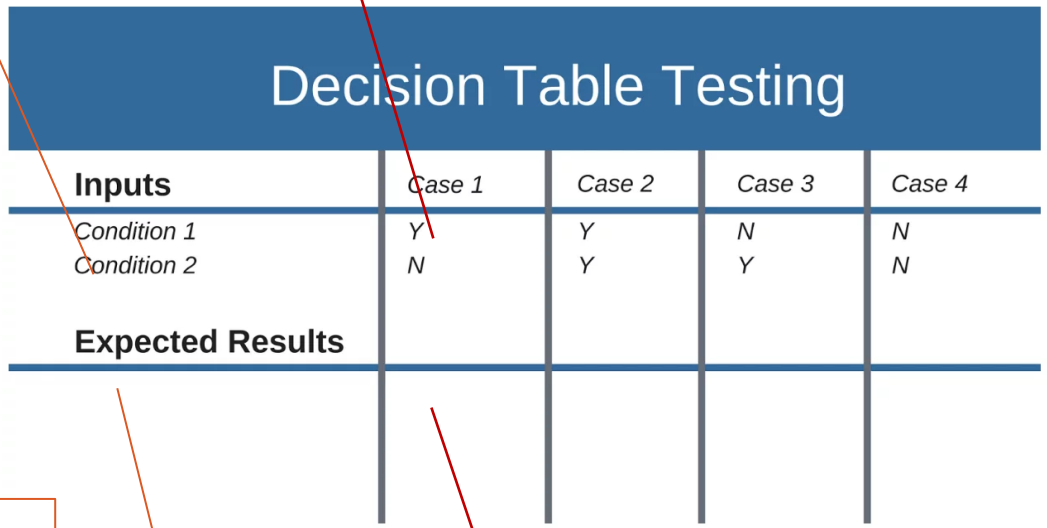
1. Identify Inputs and Outputs
2. Create the Decision Table
3. Generate Test Cases

2.3 Decision Table Testing

- Create the Decision Table
- Generate Test Cases

Condition values: any value of condition (case)

Conditions: a variable, a relationship or a statement



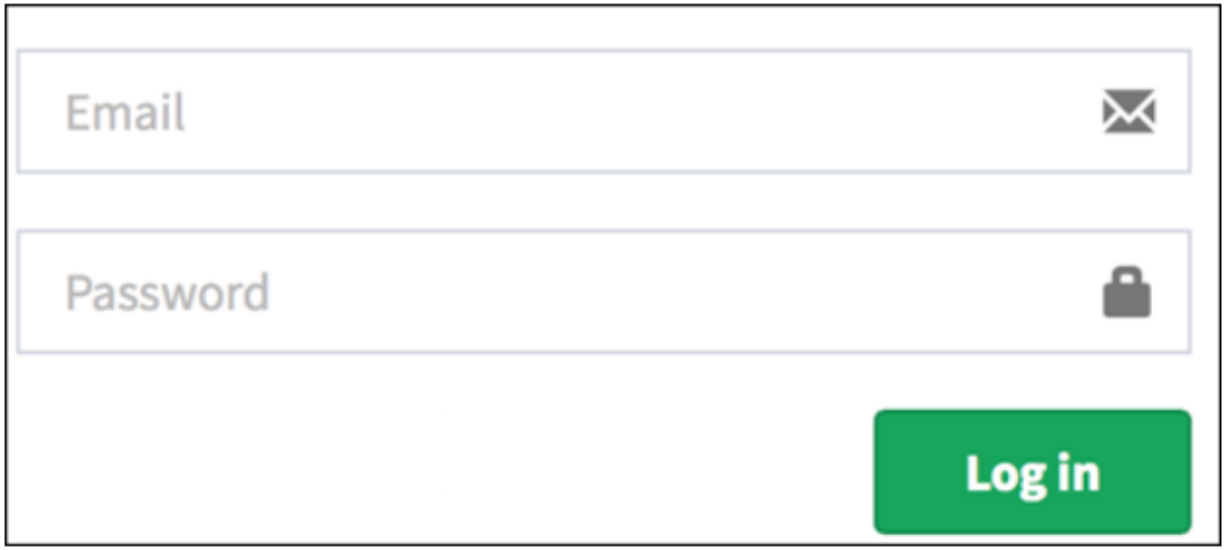
Action: a procedure or a method to implement

Act values: An action can occur depending on a combination of conditional values.

2.3 Decision Table Testing

Example 1: Login Page Functionality

Use the decision table method to test a login page with email and password inputs, ensuring it shows appropriate error messages for incorrect or blank credentials.



2.3 Decision Table Testing

Example 1: Decision Table Testing for Login Page Functionality

Step 1. Identify Inputs and Outputs

Inputs:

- Email: Valid (V), Invalid (I) , Blank (B)
- Password: Valid (V), Invalid (I) , Blank (B)

Outputs: (Expected Result)

Result 1	Result 2
Success	Login successful
Failure	Invalid email error Invalid password error Both email and password invalid error Blank email error Blank password error Both email and password blank error

2.3 Decision Table Testing

Example 1: Decision Table Testing for Login Page Functionality

Step 2&3: Create the Decision Table & Generate Test Cases

Result 1:

S: Success
F: Failure

Conditions	Email	B	B	B	I	I	I	V	V	V
	Password	B	I	V	B	I	V	B	I	V
Actions	Result	F	F	F	F	F	F	F	F	S

2.3 Decision Table Testing

Example 1: Decision Table Testing for Login Page Functionality

Step 2&3: Create the Decision Table & Generate Test Cases

Result 2:
S: Login successful
F1: Invalid email error
F2: Invalid password error
F3: Both email and password invalid error
F4: Blank email error
F5: Blank password error
F6: Both email and password blank error

Conditions	Email	B	B	B	I	I	I	V	V	V
	Password	B	I	V	B	I	V	B	I	V
Actions	Result	F4	F4	F4	F1	F1	F1	F5	F2	S

2.3 Decision Table Testing

Example 1: Decision Table Testing for Login Page Functionality

Step 2&3: Create the Decision Table & Generate Test Cases

Result 2:

S: Login successful
F1: Invalid email error
F2: Invalid password error
F4: Blank email error
F5: Blank password error

Conditions	Email	B	B	B	I	I	I	V	V	V
	Password	B	I	V	B	I	V	B	I	V
Actions	Result	F4	F4	F4	F1	F1	F1	F5	F2	S

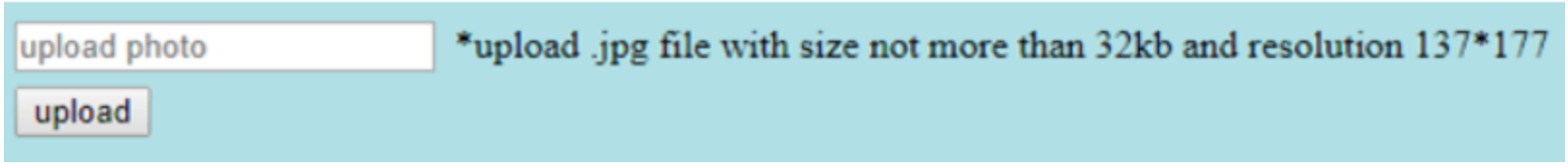
Conditions	Email	B	I	V	V	V
	Password	B	B	B	I	V
Actions	Result	F4	F1	F5	F2	S

2.3 Decision Table Testing

Example 2: Consider a dialogue box which will ask the user to upload photo with certain conditions like:

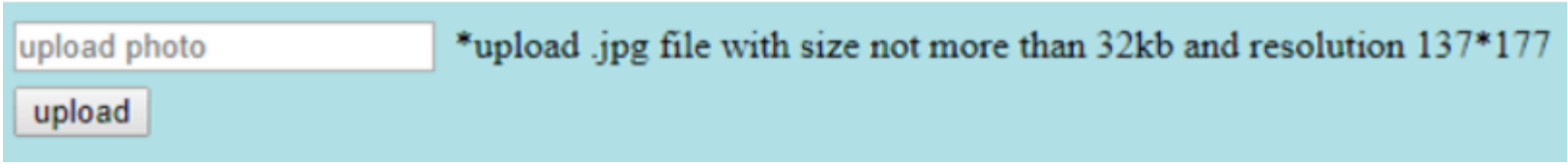
1. You can upload photo with only '.jpg' format
2. file size less than 32kb
3. resolution 137*177.

If any of the conditions fails the system will throw corresponding error message stating the issue and if all conditions are met photo will be uploaded successfully.



2.3 Decision Table Testing

Example 2:



Inputs:

- Format: .jpg, Not .jpg
- Size: <32kb, >=32kb
- Resolution: 137*177, Not 137*177

Outputs:

- Success (Photo uploaded)
- Failure (Error message)

2.3 Decision Table Testing

Example 2:

Decision table testing:

	1	2	3	4	5	6	7	8
Format	.jpg	.jpg	.jpg	.jpg	Not .jpg	Not .jpg	Not .jpg	Not .jpg
Size	<32kb	<32kb	>= 32kb	>= 32kb	<32kb	<32kb	>= 32kb	>= 32kb
Resolution	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177	137*177	Not 137*177
Outputs	S	F	F	F	F	F	F	F

2.4 State Transition Testing

State transition testing is a black-box testing technique that focuses on verifying the expected behavior of a system as it transitions between different states.

How it works:

1. Identify states
2. Identify transitions and events
3. Create test cases

2.4 State Transition Testing

Example: Bug fix process in a Bug Tracking System

Bug Dectection & Reporting

- The tester detects a bug and report it, setting the status to "Open".

Bug Evaluation

- Developers consider that if it is not the bug, they change the status to "Wont Fix", and explain the reason to the tester.
- If the tester agrees, the bug is marked as "Closed", otherwise, it is reopened as "Open".
- If it is a valid bug, developers move it to "In Progress" for fixing.

2.4 State Transition Testing

Example: Bug fix process in a Bug Tracking System

Bug fixing:

- While fixing the bug, if developers determine it's not actually a bug, they set it to "Won't Fix".
- Once fixed, the status changes to "Testing" for verification by the tester.

2.4 State Transition Testing

Example: Bug fix process in a Bug Tracking System

Verification & Closure:

- The tester checks the fix: if resolved correctly, the status is changed to “Closed”; if not, it is reopened as “Open”, and the developer must fix it again.

Bug Reappearance:

- If a closed bug reappears, the tester reopens it as “Open” and requests further fixes.

2.4 State Transition Testing

Example: Bug Tracking System

- States:

  O = Open, IP = In Progress,

  WF = Wont Fix, T = Testing, C = Closed.

- Transitions:
  - Open → In Progress, Wont Fix
  - In Progress → Wont Fix, Testing
  - Wont Fix → Open, Close
  - Testing → Open, Closed
  - Closed → Open

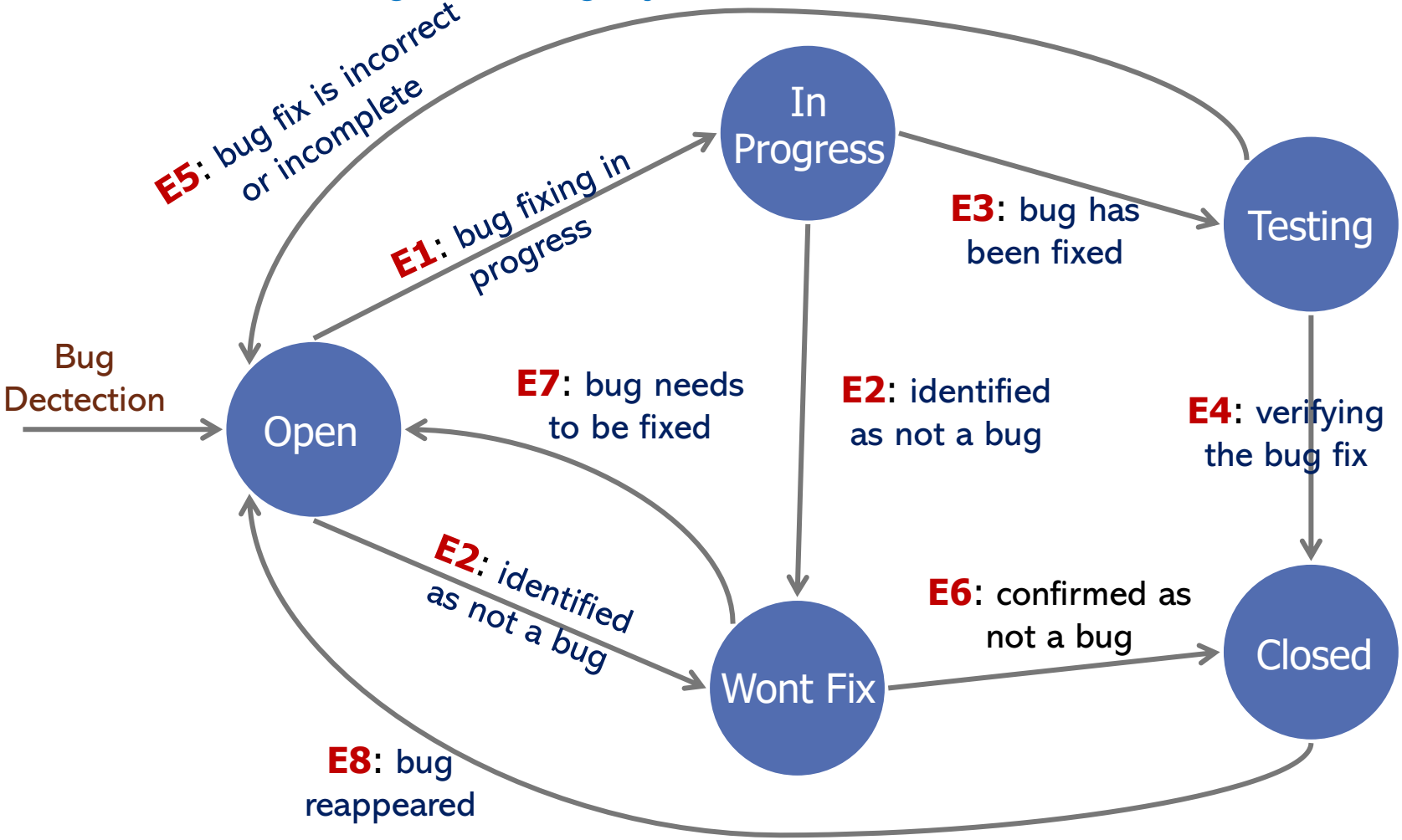
2.4 State Transition Testing

Example: Bug Tracking System

- Events:
  - E1: bug fixing in progress
  - E2: identified as not a bug
  - E3: bug has been fixed
  - E4: verifying the bug fix
  - E5: bug fix is incorrect or incomplete
  - E6: confirmed as not a bug
  - E7: bug needs to be fixed
  - E8: bug reappeared

2.4 State Transition Testing

Example: Bug Tracking System



2.4 State Transition Testing

Example: A Bug Tracking System

State transition table:

	E1	E2	E3	E4	E5	E6	E7	E8
O	IP	WF						
IP		WF	T					
T				C	O			
C								O
WF						C	O	

Based on the state table, there are 9 valid test cases: O → IP, O → WF, IP → WF, IP → T, T → C, T → O; C → O, WF → C, WF → O and some invalid cases.