Test Design and Implementation

October 2014, 2018

softserve

Agenda

- Test Design and Implementation process
- Example
- Test Case Management tools

Test Design Process

softserve

Fundamental Test Process

A test process consists of the following main groups of activities:

- **Test planning**
- **Test monitoring and control**
- **Test analysis**
- **Test design**
- **Test implementation**
- **Test execution**
- **Test completion**

softserve

Test Design and Implementation

- Test planning
- Test monitoring and control
- Test analysis
- Test design
- Test implementation
- Test execution
- Test completion

- Review and Analyze Test Basis
- Identify Test Conditions
- Design Tests using Test Design Techniques
- Design Test Environments
- Develop and Prioritize Test Cases
- Create Test Suites
- Implement Environment

softserve

Example

Driving test is an analogy for testing. We will use it to illustrate the Test Design and Implementation process.

- ✔ Test is planned and prepared in advance: routes that cover the main driving activities are planned by examiner
- ✔ The drivers under the test know the requirements of the test
- ✔ Pass/Fail criteria for driving tests are well-known
- ✔ The test is carried out to show that the driver satisfies the requirements for driving and to demonstrate that they are fit to drive

Review and Analyze Test Basis

Review and Analyze Test Basis

Identify Test Conditions

Design Tests using Test Design Techniques

Design Test Environments

Develop and Prioritize Test Cases

Create Test Suites

Implement Environment

- ✔ Review Test Basis
- ✔ Evaluate testability of the requirements and system
- ✔ Clarify requirements

Requirement to be clarified in Driving Test:

- Emergency stop: the driver must stop the car quickly, safely and without loss of control

softserve

Identify Test Conditions

Review and Analyze Test Basis

Identify Test Conditions

Design Tests using Test Design Techniques

Design Test Environments

Develop and Prioritize Test Cases

Create Test Suites

Implement Environment

✔ Define Test Conditions (as many as possible)

✔ Define test environment

Test Conditions in Driving Test:

- behavior at road junctions
- use of indicators
- ability to maneuver the car

softserve

Design Tests

Review and Analyze Test Basis

Identify Test Conditions

Design Tests using Test Design Techniques

Design Test Environments

Develop and Prioritize Test Cases

Create Test Suites

Implement Environment

✔ Define Tests for defined Conditions

Tests for ‘behavior at road junctions’
Test Conditions in Driving Test:

- T-junctions
- cross roads

softserve

Design Test Environments

- Review and Analyze Test Basis
- Identify Test Conditions
- Design Tests using Test Design Techniques
- **Design Test Environments**
- Develop and Prioritize Test Cases
- Create Test Suites
- Implement Environment

✔ Design the test environment set-up and identify any required infrastructure and tools.

Test Environment for Driving Test:

- Car (with or without additional stop pedal)

Equipment for measuring the time of response:

- stopwatch

Develop and Prioritize Test Cases

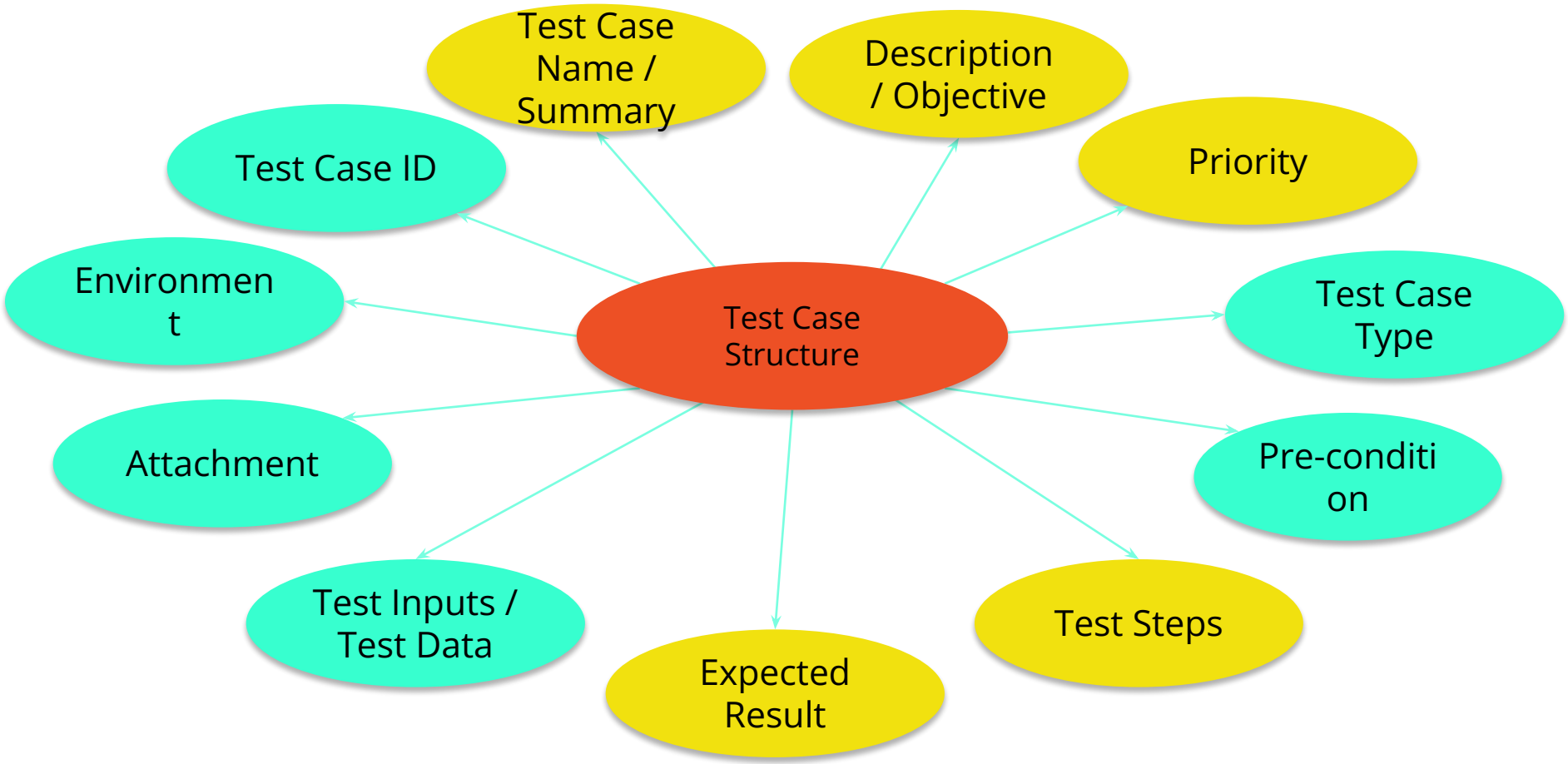
- Review and Analyze Test Basis
- Identify Test Conditions
- Design Tests using Test Design Techniques
- Design Test Environments
- **Develop and Prioritize Test Cases**
- Create Test Suites
- Implement Environment

- ✔ Develop and prioritize Test Cases
- ✔ Create Test Data for Test Cases
- ✔ Write instructions for carrying out the tests

Test Case for test condition 'junctions':

- take the route down Mayfield Road to the junction with Summer Road and ask the driver to turn left into Summer Road and then right into Green Road, expecting that the driver checks mirrors, signals and maneuvers correctly, while remaining aware of other road users.

Test Cases development



softserve

Why Test Cases?

- ✔ Testing efficiency: be ready to test once the code is ready
- ✔ Early bug detection: errors in code can be prevented before the coding is done
- ✔ Test credibility: test cases are supposed part of the deliverable to the customer
- ✔ Ability to cover all parts of the requirements
- ✔ Legal documents of testing work, in case information is needed for law suits
- ✔ Ability to track history while iterations
- ✔ Usefulness while bringing in new testers

Create Test Suites

- Review and Analyze Test Basis
- Identify Test Conditions
- Design Tests using Test Design Techniques
- Design Test Environments
- Develop and Prioritize Test Cases
- **Create Test Suites**
- Implement Environment

✔ Group Test Cases logically for Test Execution

✔ Create a Test Execution Schedule

Test Suite and Test Execution Schedule for Driving Test:

- Start the car
- Movement in forward direction
- Emergency stop

softserve

Implement Test Environments

- Review and Analyze Test Basis
- Identify Test Conditions
- Design Tests using Test Design Techniques
- Design Test Environments
- Develop and Prioritize Test Cases
- Create Test Suites
- **Implement Environment**

✔ Implement and verify Test Environment

Test Environment for Driving Test:

- Car is available
- Car is equipped by additional stop pedal
- Additional stop pedal works well

softserve

Test Design and Implementation Example

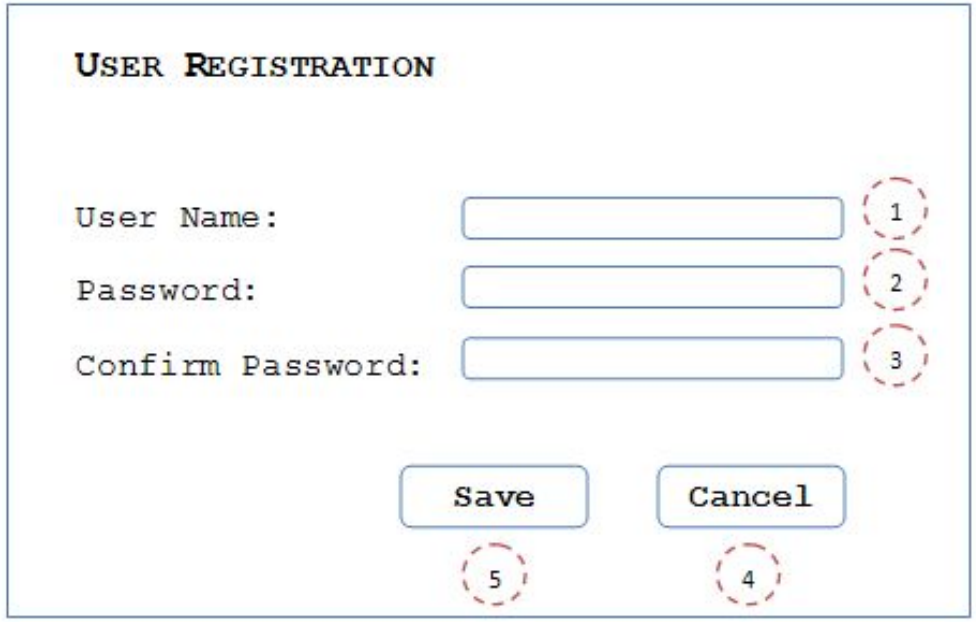
softserve

Requirements: User Registration Page

Business Value: I, as an Administrator user, should be able to create a simple user account to log in application.

Functional Requirements: 'User Registration' page should contain three fields 'User Name', 'Password', 'Confirm Password' and two buttons – 'Save' and 'Cancel'.

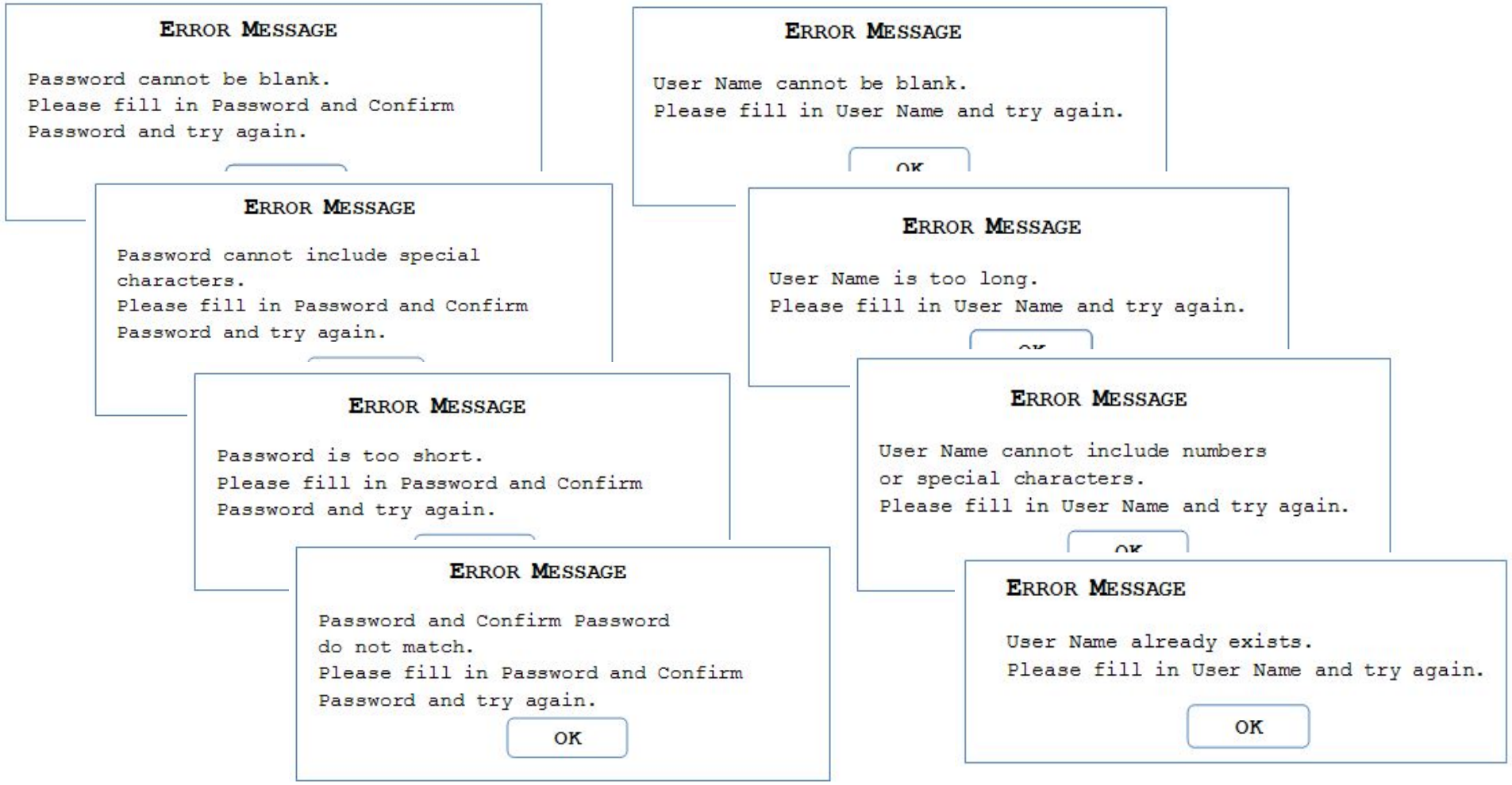
Mock up:



1. 'User Name' field is limited by 10 symbols and should contain letters of Latin alphabet only. 'User Name' field is empty by default. User Name should be unique in the system.
2. 'Password' field should be no less than 4 symbols long and should include only numbers and letters of Latin alphabet only. 'Password' field is empty by default.
3. 'Confirm Password' field should be equal to 'Password'. 'Confirm Password' field is empty by default.
4. 'Cancel' button cancels account creation and closes 'User Registration' page.
5. 'Save' button validates data entered into fields on 'User Registration' page and creates user account if entered data are correct; or shows error dialogs if validation fails. Validation should be provided in following order: User Name, Password, and Confirm Password.

softserve

Requirements: Error Messages



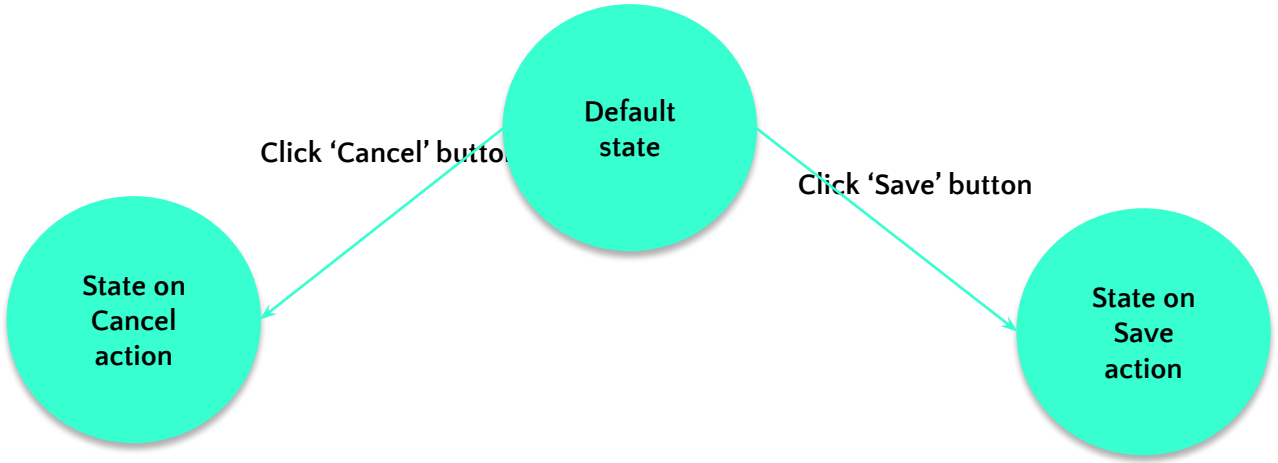
softserve

Applying State Transition Technique

‘User Name’ field is empty by default. ‘Password’ field is empty by default. ‘Confirm Password’ field is empty by default.

‘Cancel’ button cancels account creation and closes ‘User Registration’ page.

‘Save’ button creates user account if entered data are correct.

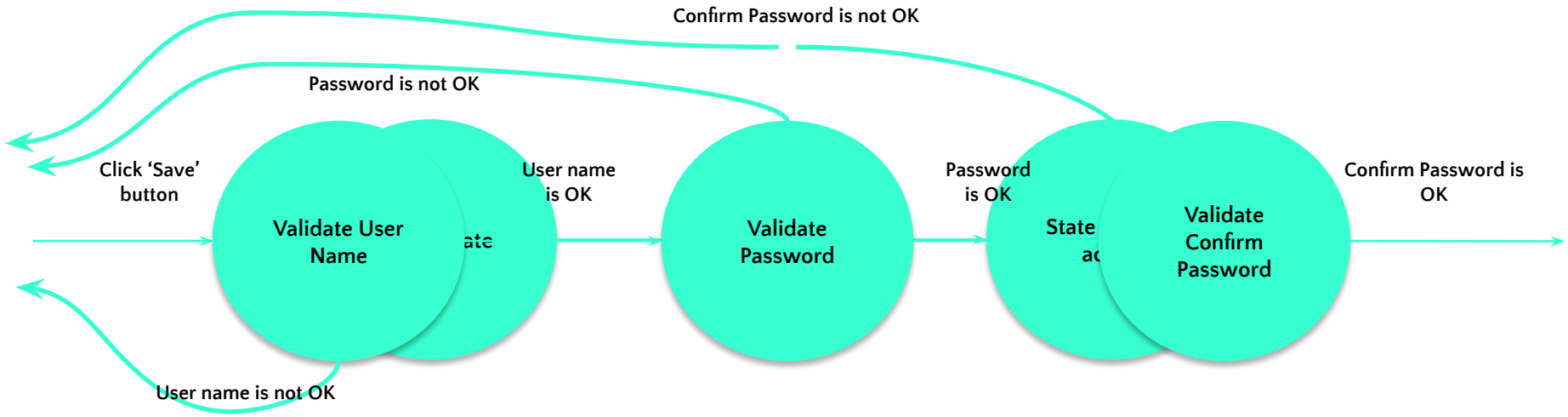


Applying State Transition Technique

Requirement	Test Name	Description
Default values	Default values on the ‘User Registration’ page	This test verifies that all fields on ‘User Registration’ page are blank by default
‘Save’ button functionality	Creating new user account and save	This test verifies that user account could be created if all fields on ‘User Registration’ page are filled with correct data; and ‘User Registration’ page is closed on save action
‘Cancel’ button functionality	Creating new user account and cancel	This test verifies that user account is not created after filling in fields on ‘User Registration’ page and canceling; and ‘User Registration’ page is closed on cancel action

Applying State Transition Technique

‘Save’ button validates data entered into fields on ‘User Registration’ page and creates user account if entered data are correct; or shows error dialogs if validation fails. Validation should be provided in following order: User Name, Password, and Confirm Password.



Applying BVA and EP Techniques

‘User Name’ field is limited by 10 symbols.

BVA: 0	1 10	11
Invalid Class	Valid Class	Invalid Class
EP: only 0	**1-10**	**11 and bigger**

‘User Name’ field should contain letters of Latin alphabet only.

Letters of Latin Alphabet	Numbers	Special Characters
Valid class	Invalid class	Invalid class
A-Z and a-z	**0-9**	**@, !, #, $, %, ^, &, *, (,), >, <, /, \, |, }, {,], [, ~, `, ‘, “, :, ;, etc.**

User Name should be unique in the system.

softserve

Test Item “User Registration”

Requirement	Test Name	Description
‘User Name’ field validation	Error dialog on saving user account with too long user name	This test verifies that error dialog appears while save action if user name length is too long: 1) boundary length – 11 characters 2) restricted length – more than 11 characters
	Error dialog on saving user account with blank ‘User Name’ field	This test verifies that error dialog appears while save action if ‘User Name’ field is blank
	Verify boundary length for user name	This test verifies that user account having user name with boundary length 1 or 10 could be created
	Error dialog on saving user account with wrong user name	This test verifies that error dialog appears while save action if ‘User Name’ field include: 1) special symbols; 2) numbers; 3) both
	Error dialog on saving already existing user account	This test verifies that error dialog appears while save action if user already exists in the system

Test Item “User Registration”

Requirement	Test Name	Description
‘Password’ field validation	Error dialog on saving user account with too short password	This test verifies that error dialog appears while save action if password length is too short: 1)boundary length – 3 characters 2)restricted length – less than 3 characters
	Error dialog on saving user account with blank ‘Password’ field	This test verifies that error dialog appears while save action if password is blank
	Verify boundary length for password	This test verifies that user account having password with boundary length 4 could be created
	Error dialog on saving user account with incorrect password	This test verifies that error dialog appears while save action if ‘Password’ field includes special symbols
‘Confirm Password’ field validation	Error dialog on saving user account with unequal password and confirm password	This test verifies that error dialog appears while save action if: 1) ’Confirm Password’ field is blank 2) password and confirm password do not match

Test Case for ‘Confirm Password’ field validation

Test Design

‘Confirm Password’ field validation	Error dialog on saving user account with unequal password and confirm password	This test verifies that error dialog appears while save action if: 1) ’Confirm Password’ field is blank 2) password and confirm password do not match

Test Case

- ✔ Example 1 – Test Data in Test Steps
- ✔ Example 2 – Test Data in Test Data field
- ✔ Example 3 – Test Data in separate document

softserve

Test case Example 1

Test Data in Test Steps

Test Case ID: 1	Test Case Name: Creating new user account and save	Status: Pass
Test Type: Functional	Author: <First and Last Names>	Creation Date: 10/17/2010
Automation: Recommended	Priority: High	Disposition: Reviewed - Completed

Objective:
This test case verifies that user account could be created if all fields on 'User Registration' page are filled with correct data; and 'User Registration' page is closed on save action.

Pre-Conditions and Setup:
Administrator user is logged to the system and 'User Registration' page is opened. If not, then run:
#1 Test Function: Open 'User Registration' page as Administrator user

User "TestUserB" do not exist in the system.

#	Test Steps	Expected Result
1	Set "TestUserB" value to 'User Name' field	
2	Set "Password1" value to 'Password' field	
3	Set "Password1" value to 'Confirm Password' field	
4	Click 'Save' button on the page	'User Registration' page is closed and just added "TestUserB" user is available in the lists of users.

Test Data:
None

Attachment(s):
None

Pros
- suitable to use when test case is a candidate for automation

Cons
- not suitable for manual testing (each time test case executes the same input values)

- hard to maintain

softserve

Test case Example 2

Test Data in Test Data field

Test Case ID: 4	Test Case Name: Verify boundary length for user name	Status: Pass
Test Type: Functional	**Author:** <First and Last Names>	**Creation Date:** 10/17/2010
Automation: Recommended	**Priority:** Medium	**Disposition:** Reviewed - Completed

Objective:
This test case verifies that user account having user name with boundary length 1 or 10 could be created.

Pre-Conditions and Setup:
Administrator user is logged to the system and 'User Registration' page is opened. If not, then run:
#1 Test Function: Open 'User Registration' page as Administrator user

Users "U" and "MyTestUser" do not exist in the system.

#	Test Steps	Expected Result
1	Set 1# value to 'User Name' field	
2	Set 2# value to 'Password' field	
3	Set 3# value to 'Confirm Password' field	
4	Click 'Save' button on the page	Just added #1 user is available in the lists of users.

Test Data:
1# "U", "MyTestUser"
2# "Password1", "Password2"
3# "Password1", "Password2"

Attachment(s):
None

Pros
- easy to maintain

- one test case can be executed with different data and you do not need to duplicate test cases

Cons
- test data field is not readable in case of a lot of data or long values

softserve

Test case Example 3

Test Data in separate document

Test Case ID: 3	**Test Case Name:** Error dialog on saving user account with wrong user name	**Status:** Fail
Test Type: Functional	**Author:** <First and Last Names>	**Creation Date:** 10/17/2010
Automation: Recommended	**Priority:** Medium	**Disposition:** Reviewed - Completed

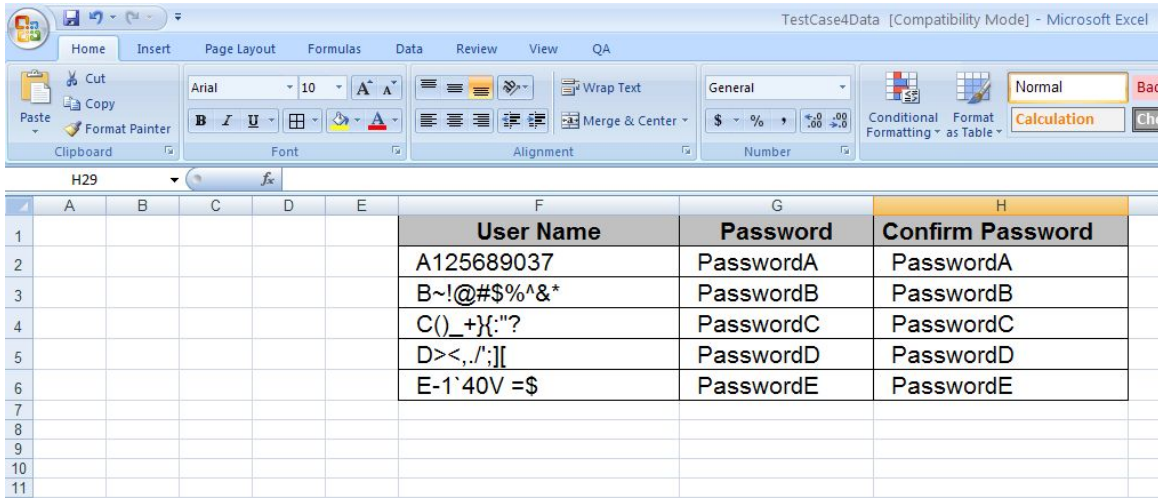
Objective:
This test case verifies that error dialog appears while save action if 'User Name' field include: 1)special symbols; 2)numbers; 3)both.

Pre-Conditions and Setup:
Administrator user is logged to the system and 'User Registration' page is opened. If not, then run:
#1 Test Function: Open 'User Registration' page as Administrator user

#	Test Steps	Expected Result
1	Set 1# value to 'User Name' field	
2	Set 2# value to 'Password' field	
3	Set 3# value to 'Confirm Password' field	
4	Click 'Save' button on the page	Error message appears in dialog window: "User Name cannot include numbers or special characters. Please fill in User Name and try again."
5	Click 'OK' button on the dialog window	

Test Data:
1# TestCase4Data.xls/Sheet1/User Name
2# TestCase4Data.xls/Sheet1/Password
3# TestCase4Data.xls/Sheet1/Confirm Password

Attachment(s):
TestCase4Data.xls



Pros
- easy to maintain data

- data in separate document are better structured

Cons
- opening separate file for each test case is time consuming

softserve

Tips and Tricks

- ✔ Write test cases for all requirements
- ✔ Write test cases with necessary detail level
- ✔ Write independent and cross-platform test cases
- ✔ Follow standard template for all test cases as well as name convention, alignment etc
- ✔ Write short test cases (up to 10-15 steps)
- ✔ Use simple English and general words
- ✔ Write test cases to quick and easy determine the expected result
- ✔ Provide test data if possible
- ✔ Write in details SQL queries (it will save time while executing)
- ✔ Add reference to bugs and requirements
- ✔ Add some notes in case you want to convey additional information
- ✔ Highlight important things, marking them in bold or assigning them color or writing in different font

Test Case Management Tools

softserve

Test Case Management Tools

Test Case Management Tool – A tool that provides support to the test management and control part of a test process.

- Microsoft Test Manager
- JIRA TCM Solution
- TestLink
- TestLog
- TestRail
- Redmine
- s
- d Silk
- Qmetry

softserve

Test Case Management Tools

Test Case Management Tool can have one or more of the following purposes depending on the context:

- ✔ Ability to create new and effectively manage existing Test Cases
- ✔ Ability to track history, Test Case executions, total run time, and estimate workload
- ✔ Ability to organize and categorize your Test Cases by Product, Component, Test Type, Test Component and Test Subcomponent
- ✔ Versioning of Test Cases
- ✔ Group Test Cases into Test Cycles
- ✔ Presence of search and filter capabilities
- ✔ Ability to link Test Cases with requirements, defects and vice versa
- ✔ Metrics gathering, reports creation, etc.

softserve

Revision History

Version	Date	Remark	Author
v.1	October, 2014		M. Harasym
v.2	October, 2018	Update according to new ISTQB Standard	V. Ryazhska

Thank you

softserve