**General Test Scenarios**

**“”””””””””””””””””””””**

* All mandatory fields should be validated and indicated by asterisk (\*) symbol
* Validation error messages should be displayed properly at correct position
* All error messages should be displayed in same CSS style (e.g. using red color)
* General confirmation messages should be displayed using CSS style other than error messages style (e.g. using green color)
* Tool tips text should be meaningful
* Dropdown fields should have first entry as blank or text like ‘Select’
* Default page sorting should be provided
* Check page sorting functionality after add/edit/delete operations on any record
* Reset button functionality should set default values for all fields
* Field labels should be standard e.g. field accepting user’s first name should be labeled properly as ‘First Name’
* Check for timeout functionality. Timeout values should be configurable. Check application behavior after operation timeout
* Check cookies used in an application
* Check if downloadable files are pointing to correct file paths
* Application crash or unavailable pages should be redirected to error page
* Check text on all pages for spelling and grammatical errors
* Check for negative numbers if allowed for numeric fields
* User should not be able to submit page twice by pressing submit button in quick succession.
* Divide by zero errors should be handled for any calculations
* Input data with first and last position blank should be handled correctly

**GUI and Usability Test Scenarios**

**“””””””””””””””””””””””””””””””**

* All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly
* Numeric values should be right justified unless specified otherwise
* Enough space should be provided between field labels, columns, rows, error messages etc.
* Scroll bar should be enabled only when necessary
* Font size, style and color for headline, description text, labels, infield data, and grid info should be standard as specified in SRS
* Description text box should be multi-line
* Disabled fields should be grayed out and user should not be able to set focus on these fields
* Upon click of any input text field, mouse arrow pointer should get changed to cursor
* User should not be able to type in drop down select lists
* Dropdown field values should be displayed in defined sort order
* Check if dropdown list options are readable and not truncated due to field size limit
* Check if correct fields are highlighted in case of errors
* Information filled by users should remain intact when there is error message on page submit. User should be able to submit the form again by correcting the errors
* Check if proper field labels are used in error messages
* Tab and Shift+Tab order should work properly
* Field specific and page level help messages should be available
* All buttons on page should be accessible by keyboard shortcuts and user should be able to perform all operations using keyboard
* Check all pages for broken images
* All pages should have title
* Confirmation messages should be displayed before performing any update or delete operation
* Hour glass should be displayed when application is busy
* Page text should be left justified
* User should be able to select only one radio option and any combination for check boxes.

**Test Scenarios for Result Grid**

**“”””””””””””””””””””””””””””**

* Page loading symbol should be displayed when it’s taking more than default time to load the result page
* Check if all search parameters are used to fetch data shown on result grid
* Total number of results should be displayed on result grid
* Search criteria used for searching should be displayed on result grid
* Result grid values should be sorted by default column.
* Result grids should include all specified columns with correct values
* Ascending and descending sorting functionality should work for columns supported with data sorting
* Result grids should be displayed with proper column and row spacing
* Pagination should be enabled when there are more results than the default result count per page
* Check for Next, Previous, First and Last page pagination functionality
* Duplicate records should not be displayed in result grid
* Check if proper symbols are used for displaying column values e.g. % symbol should be displayed for percentage calculation

**Test Scenarios for a Window**

**“”””””””””””””””””””””””””’”**

* Check if default window size is correct
* Check if child window size is correct
* Check if there is any field on page with default focus (in general, the focus should be set on first input field of the screen)
* Check if child windows are getting closed on closing parent/opener window
* If child window is opened, user should not be able to use or update any field on background or parent window
* Check window minimize, maximize and close functionality
* Check if window is re-sizable
* Check scroll bar functionality for parent and child windows

**Software Testing Checklist**

**Importance of Using Checklist for Testing:**

**“””””””””””””””””””””””””””””””””””””””**

* Maintaining a standard repository of reusable test cases for your application will ensure the most common bugs will be caught more quickly.
  + Checklist helps to quickly complete writing test cases for new versions of the application.
  + Reusing test cases help to save money on resources to write repetitive tests.
  + Important test cases will be covered always making it almost impossible to forget.
  + Testing checklist can be referred by developers to ensure most common issues are fixed in development phase itself.

**Few notes to remember:**

**“””””””””””””””””””””””**

* Execute these scenarios with different user roles e.g. admin user, guest user etc.
* For web applications these scenarios should be tested on multiple browsers like IE, FF, Chrome, and Safari with versions approved by client.
* Test with different screen resolutions like 1024 x 768, 1280 x 1024, etc.
* Application should be tested on variety of displays like LCD, CRT, Notebooks, Tablets, and Mobile phones.
* Test application on different platforms like Windows, Mac, Linux operating systems.

**Database Testing Test Scenarios**

**“””””””””””””””””””””””””””””””**

* Tables should have primary key column
* Table columns should have description information available (except for audit columns like created date, created by etc.)
* Check is values for table audit columns (like createddate, createdby, updatedate, updatedby, isdeleted, deleteddate, deletedby etc.) are populated properly

**Test Scenarios for Image Upload Functionality**

**“”””””””””””””””””””””””””””””””””””””””””””**

* Image upload progress bar should appear for large size images
* Check if cancel button functionality is working in between upload process
* Check if file selection dialog shows only supported files listed
* Check multiple images upload functionality
* Check image quality after upload. Image quality should not be changed after upload
* Check if user is able to use/view the uploaded images

**Test Scenarios for Excel Export Functionality**

**“”””””””””””””””””””””””””””””””””””””””””**

* File should get exported in proper file extension
* Check for date format if exported Excel file contains date columns
* Check number formatting for numeric or currency values. Formatting should be same as shown on page
* Exported file should have columns with proper column names
* Default page sorting should be carried in exported file as well
* Excel file data should be formatted properly with header and footer text, date, page numbers etc. values for all pages
* Check if data displayed on page and exported Excel file is same
* Check export functionality when pagination is enabled
* Check export functionality for pages containing special characters. Check if these special characters are exported properly in Excel file

**Test Scenarios for Sending Emails**

**“”””””””””””””””””””””””””””””””**

(Make sure to use dummy email addresses before executing email related tests)

* Email addresses should be validated before sending emails
* Special characters in email body template should be handled properly
* Email subject should not be blank
* Email sender name should not be blank
* Emails should be checked in different email clients like Outlook, Gmail, Hotmail, Yahoo! mail etc.
* Check send email functionality using TO, CC and BCC fields
* Check email header and footer for company logo, privacy policy and other links
* Check emails with attachments
* Check send email functionality to single, multiple or distribution list recipients
* Check if reply to email address is correct

**Performance Testing Test Scenarios**

**“”””””””””””””””””””””””””””””””””**

* Check if page load time is within acceptable range
* Check page load on slow connections
* Check response time for any action under light, normal, moderate and heavy load conditions
* Check performance of database stored procedures and triggers
* Check database query execution time
* Check for load testing of application
* Check for stress testing of application
* Check CPU and memory usage under peak load condition

**Security Testing Test Scenarios**

**“””””””””””””””””””””””””””””**

* Secure pages should use HTTPS protocol
* Page crash should not reveal application or server info. Error page should be displayed for this
* Error messages should not reveal any sensitive information
* Escape special characters in input
* All credentials should be transferred over an encrypted channel
* Check application logout functionality
* Check for SQL injection attacks
* Check for Brute Force Attacks
* Test for Denial of Service attacks
* Password should not be stored in cookies
* Cookie information should be stored in encrypted format only
* Check session cookie duration and session termination after timeout or logout
* Session tokens should be transmitted over secured channel
* Test unauthorized application access by manipulating variable values in browser address bar
* File upload functionality should use file type restrictions and also anti-virus for scanning uploaded files
* Test file’s extension handing. so that exe files are not uploaded and executed on server
* Sensitive fields like passwords and credit card information should not have auto complete enabled
* Password and other sensitive fields should be masked while typing
* Check if forgot password functionality is secured with features like security question
* Verify CAPTCHA functionality
* Check if important events are logged in log files
* Check if access privileges are implemented correctly