Springer Nature – Ela Kantor – QA Analyst

Task 1

Area under test	Search field, home page			
Considerations	In scope: Search functionality (negative and positive) and usability testing. Out of scope: Regression and load/stress testing. Search results filter. Some assumptions may not be correct (example: is there a maximum text length?) due to lack of documentation. First 2 tested subjects are the most basic/common use cases and therefore selected to be automated.			
Testing approach	Discovery/exploratory testing to identify the basic search rules. Follow common use case, as in the suggested search box. Read error messages and act according and against to instructions.			
#	Tests	Tested subject	Description	Expected result	Automated
1		Text length	Search with no input	Results are displayed	-
2			Enter minimum valid text length		Yes
3			Enter more than maximum valid test length	Page error is displayed	Yes
4			Enter more than maximum text length	Error message is displayed	-
5		Valid / invalid input	Enter valid input, including special characters and different languages	Results are displayed	Yes
6			Enter invalid text characters	'No results' message is displayed	Yes
7			Enter invalid non-text input (such as internet page)	Error message is displayed	-
8		Search suggestions	Enter valid input, longer than 2 characters	Up to 5 search suggestions appear	-
9		New search	Enter any input	New search button appears on the left of the submit search button	-
10			Click 'new search' button when appears	Button doesn't appear when	-

				search is not taking place	
11		Error message	Trigger error message upon search activity	Message is displayed clearly. Content of the message is logical	-
12		Advanced search	Enter input in all fields	Results are displayed, if applicable	-
13			Enter input in only one field		-

Task 2

Test strategy for search API engine.

First, it's important to understand at what stage of development was testing inserted and how independent are each tier (marklogic database <> core, search & content API <> Fastly webserver), these would help understand the testing boundaries. For example, it's not possible to disconnect one engine/tier without the other as both are placed in the same server.
In an ideal world, each tier is easily replaceable without effecting other tiers and unit testing was introduced at the beginning of the code.

As the team wishes to release several times a day, a rather short testing cycle should take place. However, it's recommended to act on several levels to gain confidence in test results, for key requirements:

- Unit testing – in a TDD manner: test first, then code. This would be ongoing rather than pre-release activity, but important to mention, as it would gain confidence and reduce bugs.
- End to end testing, comparing actual result to expected (known input and output). Use mock data to insure proper error testing. Accessing a testing database is needed.
- Load / performance testing. Expected requests traffic and handling errors.
- API integrity – the key and value type.
- Security testing – at least the minimum of HTTP verification.
- Postman could be a possible tool to use for some tests, as it's friendly to non-technical stake holders which might want to participate with testing.

Testing should be triggered upon request (unless scheduled releases are planned, then perhaps Jenkins could trigger automatically) and performed in staging area and only then post release testing on production.

Task 3

Area Under test	**Forgotten password feature**		
Requirements	- User settings includes a forgotten password prompt question activation (suggestion: via checkbox). Unless actively checked by the user, the option is deactivated. - Each accepted password is unique, previous password for this user could not be used.		
TBD	*Definition of a unique password*: - Is password case sensitive? - Are there new password limitations due to our wish to make is unique? (min and max length, special characters, at least one number…) *Further information is needed:* - To what extend are we saving the previous passwords? How long is it saved in our systems? - Is there a password timeout or should the password be valid for user lifetime (which is less secure)? - How is the user informed of this change? - What happens in case the user fails to answer the prompt question? Is there a limit of trials?		
Prompt question window	- Aim to help identify the user and not remember old password, as it would anyway become invalid. - Defined by the user.		
Environment	- Test user with no password history (new user). - Test user with password history (old user). - Ability to manipulate database.		

#	Tests	Tested subject	Test	Expected result
1		Default checkbox state	Create a new user. View user settings.	forgotten password prompt question checkbox is unchecked in as default.
2			Upgrade an existing user to new version. View user settings.	
3			Click 'forgot password'.	New password process has started, without a prompt question.
4		Activation. Question is prompt and answered correctly.	New user: enter any password.	All password accepted.
5			Old user: enter old password.	Error message appears.
6			Old user: enter new password.	Password accepted.

7		Parallel	Create 2 sessions with the user settings. Change password to a different one at each window and submit at the same time.	The first one is accepted, the other produces an error. In case both are truly submitted at the same time, error appears at both.
8			Create 2 sessions with the user settings. On one connect and the other forget password.	Connected user should get disconnected due to password change.
9		Lose connection.	While new password change request is being processed, connection is lost.	With no database confirmation, new password is still valid for use, and old password is still active.

Further testing is pending, as more information is required.

Springer Nature – Ela Kantor – QA Analyst