

## TASK 1

N	Repro Steps	Expected result	Actual Result	STATUS
1	fill in the following fields firstname: test; lastname: test;	Welcome test test. You credentials are: tt	Welcome test test. You credentials are: tt	OK
2	push "Submit" button without fill following fields: firstname and lastname	program should show the message, that firstname and lastname is mandatory fields	unknown: program return no message	FAIL
3	fill in the following fields firstname: 12344; lastname: test;	program shouldn't register user and should show the message that firstname should consist only with letter	Welcome 11222 test. You credentials are: 1t	FAIL
4	fill in the following fields firstname: test; lastname: 1234;	program shouldn't register user and should show the message that lastname should consist only with letter	Welcome test 1234. You credentials are: t1	FAIL
5	fill in the following fields firstname: " test" lastname: " lastname"; (fill firstname and last name which start with "space")	Welcome test lastname. You credentials are: tl	Welcome test lastname. You credentials are:	FAIL
6	firstname: sssssssssssssssssssssssssssssssssssssstetststs	there should be restriction for number of the characters in firstname and if user exceeds this amount, program should return appropriate message and should not register user	Welcome ssrr. You credentials are: sr.	FAIL
7	lastname: sssssssssssssssssssssssssssssssssssssstetststs	there should be restriction for number of the characters in lastname and if user exceeds this amount, program should return appropriate message and should not register user	Welcome test ss. You credentials are: ts	FAIL
8	fill firstname: test; and push the "submit button" without entering lastname	program shouldn't register user	Welcome test . You credentials are: t	FAIL
9	fill lastname test; and push the "submit button" without entering firstname	program shouldn't register user	Welcome test. You credentials are: t	FAIL

10	entering twice time the same user. Firstname: testtest; lastname: testtest	program should register user first time, as for second time it should give the message that the user with same credentials already exists or it should register the user with different credentials which is not exist (depend on requirments)	<b>first time:</b> Welcome testtest testtest. You credentials are: tt <b>second time:</b> Welcome testtest testtest. You credentials are: tt	<b>FAIL</b>
11	push "Submit" button multiple times	it should be imposible to submit from multiple times	request send multiple times	<b>FAIL</b>
12	fill in the following fields firstname: <b>firstname</b> <b>firstname</b> ; lastname: <b>lastname</b> .	Welcome <b>firstname</b> <b>firstname</b> <b>lastname</b> . You credentials are: fl	Welcome <b>firstname</b> <b>firstname</b> <b>lastname</b> . You credentials are: fl	<b>OK</b>
13				
14				
15				

#### Task 1.4

##### Test Cookies

- Test If the cookies are encrypted.

##### Performace testing

##### Security testing

- If the session are killed.

Task 2

	This color means attributes
	This color means components
	This color means capabilities

	User-friendly	Accurate	Secure	Portable	Multilingual	Fast	Reliable
Search	search list item number should be limited and should be used paging					Search should be return result maximum in 1 second	
Database			Sensitive data should be encrypted in database				
Payment							TCP connection should be used
Property listing	Property list should be limited and should be used paging, user also should be able to see detal						
Search filtering	Filter should contain all necessary field for searching						
Customer support					Customer should have an option to change language		
Booking		Errors during the booking should be saved and in case of error, client should not be able to make booking					

User registration			Sensitive data should be hidden. Should be used HTTPS				
Authentication			Should be used HTTPS				
Map				Customer should be able to work with map with different device			
Guides	Customer should be able to see limited number of Guids and paging should be used						
Resently seen	List should be ordered						
Review/Rating	Customer should be able to see all the review he/she made						
User settings		User should be able to properly change the user settings					
Booked hotel list	Booking hotel list should be limited and should be used paging						

N	attribute	component	capability	test cases
1	User-friendly	Search	search list item number should be limited and should be used paging	in case of search list is more than 20 item, there should be paging
2				in case of search list is less than 20 item, there should not be paging
3				search list contain more than 50 items, there should be 3 page. On click of the third page there should be only 10 item
4	Secure	Database	Sensitive data should be encrypted in database	check the user table, password should be encrypted
5				check the payment table, credit card's number should be encrypted
6				check the payment table, credit card's cvs number should be encrypted
7	Portable	Map	Customer should be able to work with map with different device	test the application map on laptop
8				test the application map on mobile phone
9				test the application map on tablet
10	User-friendly	Search filtering	Filter should contain all necessary field for searching	search the hotel by it's name. text box should be autocomplete and should
11				search the free hotel by dates, there should be possibility to choose the date with the date picker (calendar)
12				search filter's numeric fields should be dropdown, and should not give user chance to write any other value
13	User-friendly	Recently seen	List should be ordered	search one hotel with new user profile, open detail page. Go to "recently seen" page, this hotel should be only one in the list
14				search one hotel, open detail page. Go to "recently seen" page, this hotel should be the at the beginning of the list
15				search hotel, but don't open detail page. Go to "recently seen" page, this hotel should be not be in the list

Risk cluster	capabilities
Security penetration	Sensitive data in database should be encrypted
	Sensitive data should be hidden
	Should be used HTTPS
User unfriendliness	Give UX/UI design appropriate priority
	Establish customer support
	Track competition
	Search list item number should be limited and should be used paging
	Property list should be limited and should be used paging, user also should be able to see detail
Slow speed	Search should return result maximally in 1 second
	Performance tests
Narrow language selection	Localizable code
	External translators
	Authentication
	Multiple currencies support

## Task 3

Components:

### 1. Search

Positive test case – fill search textbox with following text “how to use wikiHow”, program should return search result.

negative test case - fill search textbox with following text “ssssssssssssssssssss”, there should not be any article and program should return the message “Please try another search”.

### 2. Login

Positive test case – enter **valid** password and **valid** username, user should login in system and direct to main page.

negative test case – enter **invalid** password and **valid** username, user should not login in system and will stay on the login page.

### 3. Write an article

Positive test case – try to write article with the title which is not exist, program should show following message: “Almost there! Are any of these topics the same as yours?”

negative test case – try to write article with the title which already exists, program should show following message: “That article already exists!”