

System object: Testers Hotel, http://localhost:3000/			Test-case: beforeEach Assert Index page Perform login Assert Dashboard page Assert Dashboard four block cats cats = categories	
ID 000	Pre-conditions System is accessible at localhost:3000	Post conditions None	Dependencies None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Open the browser and start the system	localhost:3000 when applicable apply: - npm run cypress:open - npm install && npm run build && npm run start	Tester Hotel application launched to login index page.	
2	Assert index page: 1 Base url 2 Site Title 3 Site header logo 4 Index page name		1 http://localhost:3000/login 2 Testers Hotel 3 Tester Hotel 4 "Login" (center below header)	
3	Inform the username	tester01		
4	Inform the password	GteteqbQQgSr88SwNExUQv2ydb7xuf8c		
5	Click the login button		Login button reads: "Login". User gains access to the system.	
6	Assert user login		User enters dashboard page. "Welcome tester01!" visible in right-end of header.	
7	Assert dashboard page: 1 Dash url 2 Site Title 3 Site header logo 4 Dashboard page name 5 Dashboard four block cats		1 localhost:3000/ 2 Testers Hotel 3 Tester Hotel 4 Tester Hotel Overview (center below header) 5 Rooms, Clients, Bills, Reservations	

System object: Testers Hotel, http://localhost:3000/			Test-case: afterEach Perform logout Assert logout	
ID 00	Pre-conditions: System is accessible at localhost:3000	Post conditions: None	Dependencies: Within any site page other than index	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Click button 'Logout'		- Logout button reads "Logout". (located far right in site header) - User exits the system and forwarded to the login page.	Logout is accessible site wide
2	Assert index page		Index page reads "Login" (center below header)	

System object: Testers Hotel, http://localhost:3000/			Test-case: confirm Rooms page Assert dashblock Rooms Enter Rooms page Assert Rooms page Exit Rooms page dashblock = dashboard page blocks	
ID 01	Pre-conditions: System is accessible at localhost:3000	Post conditions: None	Dependencies: None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Check dashblock "Rooms" has required criteria		Block contains: "Number:" and action button reads "View"	Actual digit after : is not relevant
2	Click View button		User enters Rooms page. Rooms page url reads '/rooms'	
3	Assert Rooms page criteria 1 Site header logo 2 Rooms page name 3 Action Create Room 4 Action Back		1 Tester Hotel 2 "Rooms" (center below header) 3 Button reads "Create Room", href reads '/room/new' 4 Button reads "Back"	
4	Click Button "Back"		User returns to Dashboard page	
5	Assert Dashboard page		url reads 'localhost:3000/' page title reads 'Tester Hotel Overview'	center below header

System object: Testers Hotel, http://localhost:3000/			Test-case: confirm Clients page Assert dashblock Clients Enter Clientss page Assert Clients page Exit Clients page dashblock = dashboard page blocks	
ID 02	Pre-conditions: System is accessible at localhost:3000	Post conditions: None	Dependencies: None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Check dashblock "Clients" has required criteria		Block contains: "Number:" and action button reads "View"	Actual digit after : is not relevant
2	Click View button		User enters Clients page. Clients page url reads '/clients'	
3	Assert Clients page criteria 1 Site header logo 2 Clients page name 3 Action Create Client 4 Action Back		1 Tester Hotel 2 "Clients" (center below header) 3 Button reads "Create Client", href reads '/client/new' 4 Button reads "Back"	
4	Click Button "Back"		User returns to Dashboard page	
5	Assert Dashboard page		url reads 'localhost:3000/' page title reads 'Tester Hotel Overview'	

System object: Testers Hotel, http://localhost:3000/			Test-case: confirm Bills page Assert dashblock Bills Enter Bills page Assert Bills page Exit Bills page dashblock = dashboard page blocks	
ID 03	Pre-conditions: System is accessible at localhost:3000	Post conditions: None	Dependencies: None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Check dashblock "Bills" has required criteria		Block contains: "Total:" , "Paid:". action button reads "View"	Actual digit after : is not relevant
2	Click View button		User enters Bills page. Bills page url reads '/bills'	
3	Assert Bills page criteria 1 Site header logo 2 Bills page name 3 Action Create Bill 4 Action Back		1 Tester Hotel 2 "Bills" (center below header) 3 Button reads "Create Bill", href reads '/bill/new' 4 Button reads "Back"	
4	Click Button "Back"		User returns to Dashboard page	
5	Assert Dashboard page		url reads 'localhost:3000/' page title reads 'Tester Hotel Overview'	

System object: Testers Hotel , http://localhost:3000/			Test-case: confirm Reservations page Assert dashblock Reservation Enter Reservation page Assert Reservation page Exit Reservation page dashblock = dashboard page blocks	
ID 04	Pre-conditions: System is accessible at localhost:3000	Post conditions: None	Dependencies: None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Check dashblock "Reservations" has required criteria		Block contains: "Total:" , "Current:" action button reads "View"	Actual digit after : is not relevant
2	Click View button		User enters Reservations page. Reservations page url reads '/reservations'	
3	Assert Reservations page criteria 1 Site header logo 2 Reservations page name 3 Action Create Reservation 4 Action Back		1 Tester Hotel 2 "Reservations" (center below header) 3 Button reads "Create Reservation", href reads '/reservation/new' 4 Button reads "Back"	
4	Click Button "Back"		User returns to Dashboard page	
5	Assert Dashboard page		url reads 'localhost:3000/' page title reads 'Tester Hotel Overview'	

System object: Testers Hotel, http://localhost:3000/			Test-case: Create new client Assert New Client page Assert empty form field error validations Assert new client form works Assert new client is saved & visible	
ID 05	Pre-conditions: System is accessible at localhost:3000	Post conditions: Create new client and click button Save for client card to be visible in clients list.	Dependencies: None	Tester SJ
Step	Test step	Test data	Expected result	Notes
1	Assert new client page			
	1 Start at clients page 2 Click button "Create Client" 3 New client url 4 Page name, center below header 5 action Back	1 (localhost:3000/clients)	2 User enters New Client page 3 page url reads '/client/new' 4 Reads "New Client" 5 Button reads "Back" href reads '/clients'	
2	Assert empty form field error handling			
	Click button "Save"	Form fields are left empty	(array) error messg length = 3 Error message for each empty form field: "Name must be set" "Email must be set" "Telephone must be set"	
3	Assert form labels and type data into each corresponding input field		(array) label length = 3 Label values: Name, Email, Telephone	
	Name field	Samy J.		didn't check attr type text
	Email field	samyj@gmail.com	Attribute type = email	
	Telephone field	+1 (415) 444-4501	Attribute type = text →	type number ?
	Assert step 2 error message handling		All 3 error messages disappear one by one on the page after each field has correct formatted input.	
4	Click button "Save"		User is directed to Clients page	
5	Clients page url Page name, center below header		Reads '/clients' Reads 'Clients'	
6	Assert new client card and correct data		New client card is visible. New card is placed last in clients list.	Suggestion: client list be descending order: newest to oldest instead of current descending order.
	Client card initials icon		Uppercase first & last client initials : SJ	
	Client card data		Reads matching client info input from step 3 . Starting with Name, ending with Telephone.	
End				