Remember that there is a difference between several scenarios:

* An element not being present at all in the DOM.
* An element being present in the DOM but not visible.
* An element being present in the DOM but not enabled. (i.e. clickable)

My guess is that if some of the page is being displayed with javascript, the elements are already present in the browser DOM, but are not visible. The implicit wait only waits for an element to appear in the DOM, so it returns immediately, but when you try to interact with the element you get a NoSuchElementException. You could test this hypothesis by writing a helper method that explicits waits for an element to be be visible or clickable.

Some examples (in Java):

public WebElement getWhenVisible(By locator, int timeout) {

WebElement element = null;

WebDriverWait wait = new WebDriverWait(driver, timeout);

element = wait.until(ExpectedConditions.visibilityOfElementLocated(locator));

return element;

}

public void clickWhenReady(By locator, int timeout) {

WebDriverWait wait = new WebDriverWait(driver, timeout);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(locator));

element.click();

}