Sure, here are some possible test cases for a login page in a library management system. In these examples, I'll assume you're using a web-based application and the Selenium WebDriver library for automation testing in Python. Make sure you have Selenium installed (**pip install selenium**) and the appropriate web driver (like ChromeDriver) set up.

**Test Cases for Login Page:**

Test Case 1: Valid Credentials

* **Input:** Valid username and password
* **Expected Output:** Redirect to the dashboard page
* **Automation Test Code:**

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Initialize the WebDriver

driver = webdriver.Chrome(executable\_path='path/to/chromedriver')

# Open the login page

driver.get("http://example.com/login")

# Find username and password fields and enter valid credentials

username = driver.find\_element\_by\_name("username")

password = driver.find\_element\_by\_name("password")

username.send\_keys("valid\_username")

password.send\_keys("valid\_password")

# Submit the form

password.send\_keys(Keys.RETURN)

# Verify if redirected to the dashboard page

assert "Dashboard" in driver.title

# Close the browser

driver.close()

Test Case 2: Invalid Credentials

* **Input:** Invalid username or password
* **Expected Output:** Error message displayed
* **Automation Test Code:**

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Initialize the WebDriver

driver = webdriver.Chrome(executable\_path='path/to/chromedriver')

# Open the login page

driver.get("http://example.com/login")

# Find username and password fields and enter invalid credentials

username = driver.find\_element\_by\_name("username")

password = driver.find\_element\_by\_name("password")

username.send\_keys("invalid\_username")

password.send\_keys("invalid\_password")

# Submit the form

password.send\_keys(Keys.RETURN)

# Verify if error message is displayed

assert "Invalid credentials" in driver.page\_source

# Close the browser

driver.close()

Test Case 3: Empty Fields

* **Input:** Empty username and password fields
* **Expected Output:** Error message displayed
* **Automation Test Code:**

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

# Initialize the WebDriver

driver = webdriver.Chrome(executable\_path='path/to/chromedriver')

# Open the login page

driver.get("http://example.com/login")

# Find username and password fields and leave them empty

username = driver.find\_element\_by\_name("username")

password = driver.find\_element\_by\_name("password")

# Submit the form

password.send\_keys(Keys.RETURN)

# Verify if error message is displayed

assert "Please enter username and password" in driver.page\_source

# Close the browser

driver.close()

Please replace **"http://example.com/login"**, **"valid\_username"**, and **"valid\_password"** with the actual URL of your login page, valid username, and password for testing.

Make sure to handle your specific error messages and element locators in the test code according to your application's implementation. Also, you can use testing frameworks like Pytest to enhance your test cases and reports.