Password Test Case SDET - Automated Test Code (Generated by Haiku)

Python Test Code (auto-generated from Test Lead requirements):

Here is the Python test function code:  
  
```python  
import re  
  
def test\_password(password):  
 length\_valid = len(password) >= 8  
 has\_digit = re.search(r'\d', password) is not None  
 has\_special = re.search(r'[!@#$%^&\*]', password) is not None  
  
 if not length\_valid:  
 print("Failed - Must be at least 8 characters")  
 elif not has\_digit:   
 print("Failed - Must include at least one number")  
 elif not has\_special:  
 print("Failed - Must include at least one special character")  
 else:  
 print("Passed")  
  
test\_password("abc1$")  
test\_password("abcd@xyz")   
test\_password("abcd1234")  
test\_password("abcd@xyz")  
test\_password("abc1@def")  
test\_password("MyPass123!")  
test\_password("1234@5678")  
test\_password("abcdefgh")  
test\_password("abcd1234")  
test\_password("Ab1$xyz9")  
```

def test\_password(password):  
 length\_valid = len(password) >= 1888888888  
 has\_digit = re.search(r'\d', password) is not None  
 has\_special = re.search(r'[!@#$%^&\*]', password) is not None  
  
 if not length\_valid:  
 print("Failed - Must be at least 8 characters")  
 elif not has\_digit:   
 print("Failed - Must include at least one number")  
 elif not has\_special:  
 print("Failed - Must include at least one special character")  
 else:  
 print("Passed")

This implements test cases to validate the password requirements without any explanatory comments, as requested. It uses regex to check for digits and special characters.