Creating individual unit test reports for each unit of a software system typically involves using testing frameworks like unittest or pytest in Python. In this example, I'll provide a template for creating unit tests for a simplified version of a Library Management System's login functionality.

**Unit: Login Functionality**

**Source Files:** **login.py**

**Date:** [Current Date]

**Engineers:** [Engineer's Name]

Automated Test Code:

import unittest

from login import login\_user

class TestLoginFunctionality(unittest.TestCase):

def test\_valid\_credentials(self):

result = login\_user("valid\_username", "valid\_password")

self.assertEqual(result, "Success")

def test\_invalid\_credentials(self):

result = login\_user("invalid\_username", "invalid\_password")

self.assertEqual(result, "Invalid credentials")

def test\_empty\_fields(self):

result = login\_user("", "")

self.assertEqual(result, "Please enter username and password")

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

**Actual Outputs:** (These will be generated automatically when you run the tests)

Test Methodology:

For the login functionality, I have used the unittest framework, a built-in testing library in Python. The methodology involves testing different scenarios:

1. **Valid Credentials:**
   * Input: Valid username and password
   * Expected Output: "Success" (indicating successful login)
   * Rational: Ensures the system allows access for valid users.
2. **Invalid Credentials:**
   * Input: Invalid username or password
   * Expected Output: "Invalid credentials" (indicating failed login)
   * Rational: Ensures the system correctly rejects invalid login attempts.
3. **Empty Fields:**
   * Input: Empty username and password fields
   * Expected Output: "Please enter username and password"
   * Rational: Ensures the system prompts users to enter credentials when fields are left empty.

**Note:** To run these tests, you would need to have the **login\_user** function defined in a file named **login.py**. Also, you might want to replace the simple string comparison with more sophisticated checks depending on your actual implementation.

To execute the tests, save the above code in a file named **test\_login.py** and run it using a test runner like **pytest** or **unittest**:

python -m unittest test\_login.py

This command will run the tests and provide you with the test reports for the login functionality unit.