

.....4.02.2023.....

data oddania projektu

STUDIA PODYPŁOMOWE „Tester oprogramowania”

Wyższa Szkoła Bankowa w Warszawie

MINI PROJEKT – AUTOMATYZACJA TESTÓW DLA APLIKACJI MOBILNEJ

Cel projektu: Automatyzacja przypadku testowego / przypadków testowych dla aplikacji mobilnej android lub ios

Sprawdził:

mgr inż. Mateusz Brzeziński

Opracowała:

Anna Świeżawska

Nazwa testowanej aplikacji: Android_Demo_App.apk
„KWADemo”

I. Przypadek testowy / Przypadki Testowe

ID:001.

Tytuł: Poprawne logowanie admina do aplikacji

Środowisko:

System operacyjny komputera: Windows 10

Android Studio: 2021.1.1. Patch2

Testy z wykorzystaniem emulatora – Google Pixel 4 XL

Wersja system operacyjnego mobilnego: Android 12.0

| Java | Pycharm | Python | Android studio | Appium |
|--------|------------------------------------|--------|--|--------|
| 17.0.6 | 2022.3.2 (Community Edition) | 3.10.9 | 11.0.15+0- b2043.56- 8887301 amd64 | 1.22.2 |

Warunek wstępny: Uruchomiona aplikacja. Użytkownik nie jest zalogowany.

Kroki:

1. Open log in menu (otwórz menu z logowaniem i kliknij w log in przycik)
2. Enter valid email (wprowadź poprawny email)
3. Enter password (wprowadź hasło)
4. Click on login button (kliknij w login przycisk)
5. Enter Admin (wpisz imię admina)
6. Click submit button (kliknij w zatwierdź przycisk)
7. Check if User is logged (sprawdź czy użytkownik jest zalogowany)

Oczekiwany rezultat:

Użytkownik zostaje poprawnie zalogowany.

ID:002

Tytuł: Błędne logowanie admina do aplikacji

Środowisko:

System operacyjny komputera: Windows 10

Android Studio: 2021.1.1. Patch2

Testy z wykorzystaniem emulatora – Google Pixel 4 XL

Wersja system operacyjnego mobilnego: Android 12.0

| Java | Pycharm | Python | Android studio | Appium |
|--------|------------------------------------|--------|----------------------------|--------|
| 17.0.6 | 2022.3.2 (Community Edition) | 3.10.9 | 17.0.5+1- b653.25 amd64 | 1.22.2 |

Warunek wstępny: Uruchomiona aplikacja. Użytkownik nie jest zalogowany.

Kroki:

1. Open log in menu (otwórz menu z logowaniem i kliknij w log in przycisk)
2. Enter invalid email (wprowadź niepoprawny email)
3. Enter password (wprowadź hasło)
4. Click on login button (kliknij w login przycisk)
5. Check if „Wrong Credentials” is displayed on the screen (sprawdź czy tekst “Wrong Credentials” jest wyświetlony na ekranie)
6. Check if “username: admin@gmail” is displayed on the screen (sprawdź czy tekst “username: admin@gmail” jest wyświetlony na ekranie)
7. Check if “password: admin 123” is displayed on the screen (sprawdź czy tekst “password: admin 123” jest wyświetlony na ekranie)

Oczekiwany rezultat:

Użytkownik nie zostaje zalogowany.

ID:003

Tytuł: Sprawdzenie poprawności działania pola „Enter value”

Środowisko:

System operacyjny komputera: Windows 10

Android Studio: 2021.1.1. Patch2

Testy z wykorzystaniem emulatora – Google Pixel 4 XL

Wersja system operacyjnego mobilnego: Android 12.0

| Java | Pycharm | Python | Android studio | Appium |
|--------|---------------------------------|--------|------------------------|--------|
| 17.0.6 | 2022.3.2 (Community Edition) | 3.10.9 | 17.0.5+1-b653.25 amd64 | 1.22.2 |

Warunek wstępny: Uruchomiona aplikacja.

Kroki:

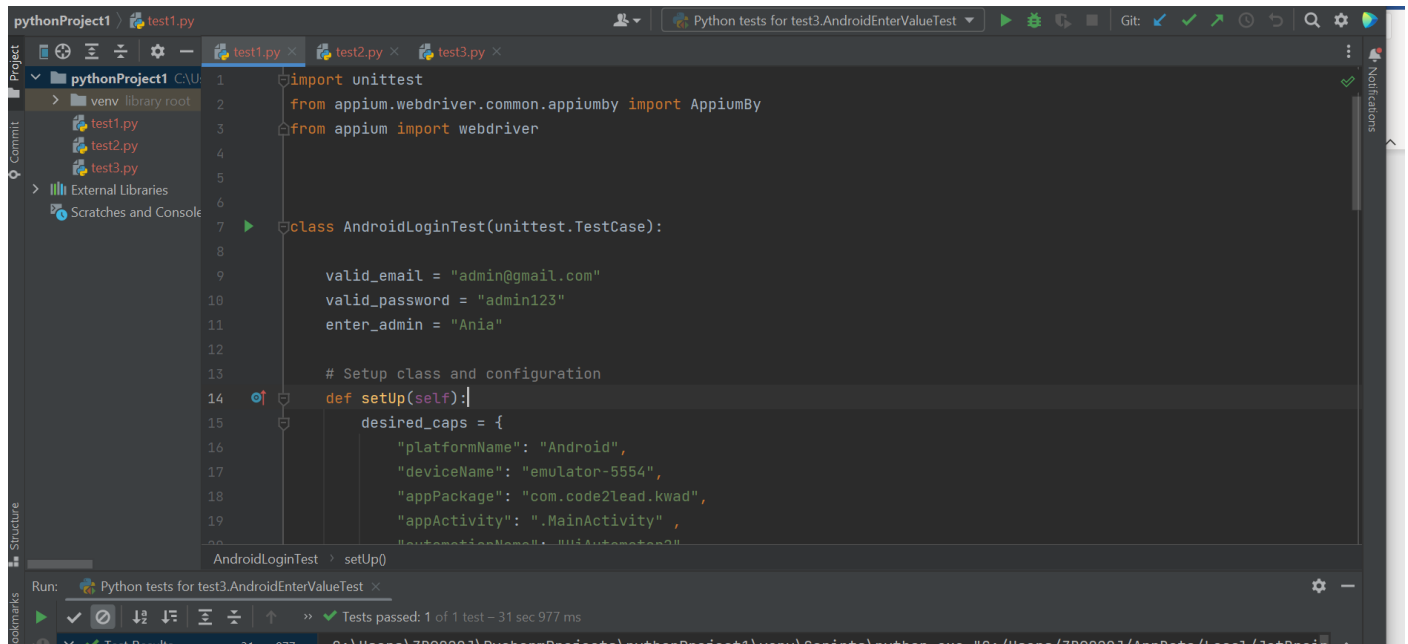
1. On my page: check if my page is loaded (na mojej stronie: sprawdź czy strona się załadowała)
2. Click on “Enter some value” field (kliknij w pole “Enter some value”)
3. Enter “hello world!” in the “Enter some value” field (napisz “hello world” w polu “Enter some value”)
4. Click “Submit” button (kliknij w “Submit” przycisk)
5. Check if values “hello world” are displayed (sprawdź czy wprowadzone wartości są wyświetlone na stronie „hello world!”)

Oczekiwany rezultat:

Pole działa poprawnie. Po wprowadzeniu wartości do pola, są one wyświetlane na stronie ekranu.

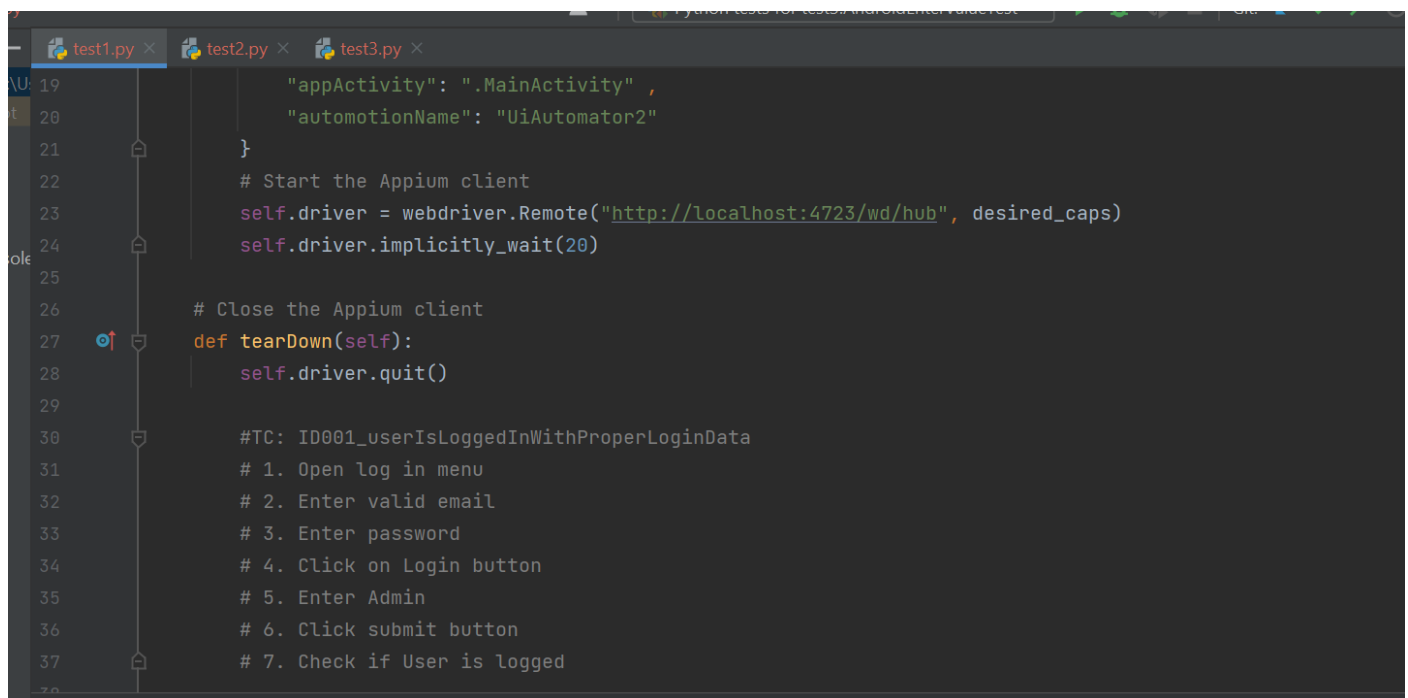
II. Automatyzacja Przypadków Testowych

ID001



The screenshot shows an IDE with a project named 'pythonProject1'. The file explorer on the left shows a 'venv' directory and three test files: 'test1.py', 'test2.py', and 'test3.py'. The main editor displays the code for 'test1.py', specifically the 'setUp' method of the 'AndroidLoginTest' class. The code imports 'unittest' and 'AppiumBy' from 'appium.webdriver.common.appiumby', and 'webdriver' from 'appium'. It defines a class 'AndroidLoginTest' that inherits from 'unittest.TestCase'. Inside the class, it sets 'valid_email' to 'admin@gmail.com', 'valid_password' to 'admin123', and 'enter_admin' to 'Ania'. The 'setUp' method defines 'desired_caps' with the following values: 'platformName': 'Android', 'deviceName': 'emulator-5554', 'appPackage': 'com.code2lead.kwad', 'appActivity': '.MainActivity', and 'automationName': 'UiAutomator2'. The status bar at the bottom indicates that the tests passed: 'Tests passed: 1 of 1 test - 31 sec 977 ms'.

```
1 import unittest
2 from appium.webdriver.common.appiumby import AppiumBy
3 from appium import webdriver
4
5
6
7 class AndroidLoginTest(unittest.TestCase):
8
9     valid_email = "admin@gmail.com"
10    valid_password = "admin123"
11    enter_admin = "Ania"
12
13    # Setup class and configuration
14    def setUp(self):
15        desired_caps = {
16            "platformName": "Android",
17            "deviceName": "emulator-5554",
18            "appPackage": "com.code2lead.kwad",
19            "appActivity": ".MainActivity",
20            "automationName": "UiAutomator2"
21        }
```

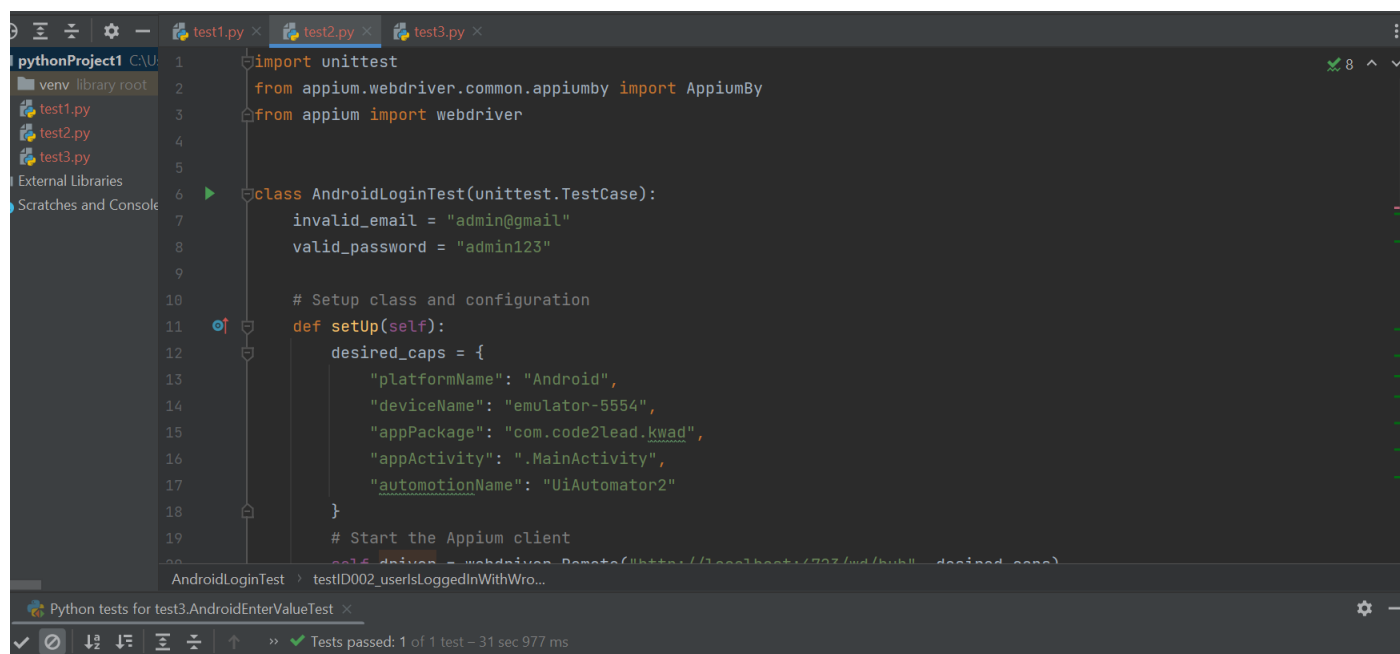


The screenshot shows the continuation of the 'test1.py' file in the IDE. It displays the 'tearDown' method and a list of test steps. The 'tearDown' method calls 'self.driver.quit()'. Below it, a comment indicates the test case: '#TC: ID001_userIsLoggedInWithProperLoginData'. A list of seven steps follows: 1. Open log in menu, 2. Enter valid email, 3. Enter password, 4. Click on Login button, 5. Enter Admin, 6. Click submit button, and 7. Check if User is logged.

```
19     "appActivity": ".MainActivity" ,
20     "automationName": "UiAutomator2"
21 }
22 # Start the Appium client
23 self.driver = webdriver.Remote("http://localhost:4723/wd/hub", desired_caps)
24 self.driver.implicitly_wait(20)
25
26 # Close the Appium client
27 def tearDown(self):
28     self.driver.quit()
29
30 #TC: ID001_userIsLoggedInWithProperLoginData
31 # 1. Open log in menu
32 # 2. Enter valid email
33 # 3. Enter password
34 # 4. Click on Login button
35 # 5. Enter Admin
36 # 6. Click submit button
37 # 7. Check if User is logged
```

```
test1.py x test2.py x test3.py x
37 # 7. Check if User is logged
38
39 def testID001_userIsLoggedInWithProperLoginData(self):
40     # 1. Open Log in menu
41     el = self.driver.find_element(AppiumBy.ACCESSIBILITY_ID, "Btn6")
42     # Check if element is displayed
43     self.assertTrue(el.is_displayed())
44     el.click()
45     # 2. Enter valid email
46     email_field = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Et4")
47     email_field.send_keys(self.valid_email)
48     # 3. Enter password
49     password_field = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Et5")
50     password_field.send_keys(self.valid_password)
51     # 4. Click on Login button
52     login_button = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Btn3")
53     login_button.is_displayed()
54     login_button.click()
55     # 5. Enter Admin: Ania
56     _enter_admin = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Edt_admin")
57     _enter_admin.send_keys(self.enter_admin)
58
59 def setUp(self):
60     self.driver = webdriver.Chrome()
61     self.driver.get("http://10.0.2.15:8080")
62     self.driver.implicitly_wait(10)
63     self.valid_email = "admin@code2lead.com"
64     self.valid_password = "Admin@123"
65     self.enter_admin = "Ania"
66
67 if __name__ == '__main__':
68     unittest.main()
```

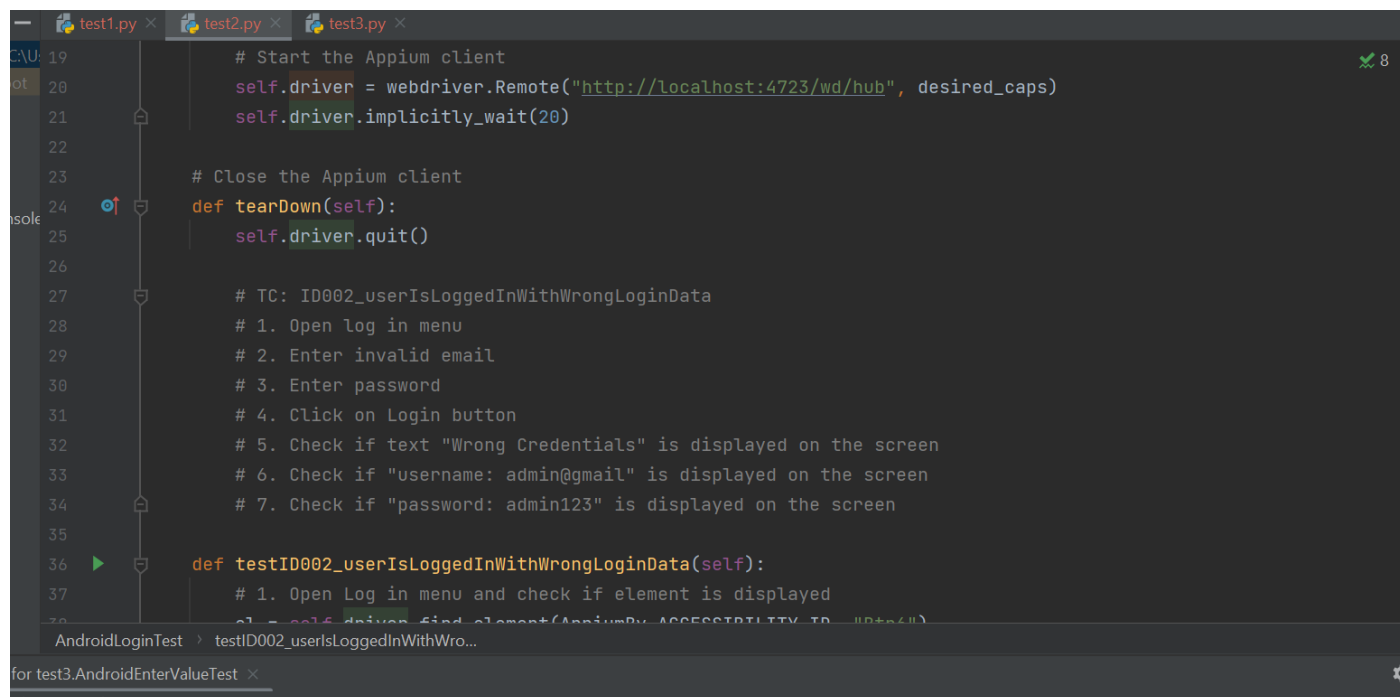
```
Python tests for test3.AndroidEnterValueTest
test1.py x test2.py x test3.py x
53 login_button.is_displayed()
54 login_button.click()
55 # 5. Enter Admin: Ania
56 _enter_admin = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Edt_admin")
57 _enter_admin.send_keys(self.enter_admin)
58 # 6. Click submit button
59 submit_button = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Btn_admin_sub")
60 submit_button.is_displayed()
61 submit_button.click()
62 # 7. Check if User is logged
63 user_logged = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv_admin")
64 self.assertTrue(user_logged.is_displayed())
65 user_logged.click()
66
67
68
69 if __name__ == '__main__':
70     unittest.main()
```



The screenshot shows an IDE with three tabs: test1.py, test2.py, and test3.py. The test1.py tab is active, displaying the following code:

```
1 import unittest
2 from appium.webdriver.common.appiumby import AppiumBy
3 from appium import webdriver
4
5
6 class AndroidLoginTest(unittest.TestCase):
7     invalid_email = "admin@gmail"
8     valid_password = "admin123"
9
10    # Setup class and configuration
11    def setUp(self):
12        desired_caps = {
13            "platformName": "Android",
14            "deviceName": "emulator-5554",
15            "appPackage": "com.code2lead.kwad",
16            "appActivity": ".MainActivity",
17            "automationName": "UiAutomator2"
18        }
19        # Start the Appium client
```

The bottom status bar indicates: Python tests for test3.AndroidEnterValueTest x Tests passed: 1 of 1 test - 31 sec 977 ms.



The screenshot shows the same IDE with the test2.py tab active, displaying the following code:

```
19 # Start the Appium client
20 self.driver = webdriver.Remote("http://localhost:4723/wd/hub", desired_caps)
21 self.driver.implicitly_wait(20)
22
23 # Close the Appium client
24 def tearDown(self):
25     self.driver.quit()
26
27 # TC: ID002_userIsLoggedInWithWrongLoginData
28 # 1. Open log in menu
29 # 2. Enter invalid email
30 # 3. Enter password
31 # 4. Click on Login button
32 # 5. Check if text "Wrong Credentials" is displayed on the screen
33 # 6. Check if "username: admin@gmail" is displayed on the screen
34 # 7. Check if "password: admin123" is displayed on the screen
35
36 def testID002_userIsLoggedInWithWrongLoginData(self):
37     # 1. Open Log in menu and check if element is displayed
38     el = self.driver.find_element(AppiumBy.ACCESSIBILITY_ID, "ID002")
```

The bottom status bar indicates: Python tests for test3.AndroidEnterValueTest x.

```
test1.py × test2.py × test3.py ×
37 # 1. Open Log in menu and check if element is displayed
38 el = self.driver.find_element(AppiumBy.ACCESSIBILITY_ID, "Btn6")
39 self.assertTrue(el.is_displayed())
40 el.click()
41 # 2. Enter invalid email
42 email_field = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Et4")
43 email_field.send_keys(self.invalid_email)
44 # 3. Enter password
45 password_field = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Et5")
46 password_field.send_keys(self.valid_password)
47 # 4. Click on Login button
48 login_button = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Btn3")
49 login_button.is_displayed()
50 login_button.click()
51 # 5. Check if text "Wrong Credentials" is displayed on the screen
52 wrong_credentials = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv8")
53 wrong_credentials.click()
54 self.assertTrue(wrong_credentials.is_displayed())
55 # 6. Check if text "username: admin@gmail" is displayed on the screen
56 displayed_text = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv3")
57 displayed_text.click()
58 self.assertTrue(displayed_text.is_displayed())
59 # 7. Check if text "password: admin123" is displayed on the screen
60 displayed_password = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv4")
61 self.assertTrue(displayed_password.is_displayed())
62 displayed_password.click()
63
64
65 if __name__ == '__main__':
66     unittest.main()
67
```

AndroidLoginTest > testID002_usersLoggedInWithWro...

test3.AndroidEnterValueTest ×

Tests passed: 1 of 1 test - 31 sec 977 ms

```
test1.py × test2.py × test3.py ×
53 wrong_credentials.click()
54 self.assertTrue(wrong_credentials.is_displayed())
55 # 6. Check if text "username: admin@gmail" is displayed on the screen
56 displayed_text = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv3")
57 displayed_text.click()
58 self.assertTrue(displayed_text.is_displayed())
59 # 7. Check if text "password: admin123" is displayed on the screen
60 displayed_password = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv4")
61 self.assertTrue(displayed_password.is_displayed())
62 displayed_password.click()
63
64
65 if __name__ == '__main__':
66     unittest.main()
67
```

ID003

The screenshot displays an IDE with a Python test file named `test3.py` open. The code defines a test class `AndroidEnterValueTest` that inherits from `unittest.TestCase`. The class has a `setUp` method that configures the Appium client with specific capabilities for an Android emulator. The test class is named `AndroidEnterValueTest`.

```
1 import unittest
2 from appium.webdriver.common.appiumby import AppiumBy
3 from appium import webdriver
4
5
6 class AndroidEnterValueTest(unittest.TestCase):
7     enter_value = "hello world!"
8
9     # Setup class and configuration
10    def setUp(self):
11        desired_caps = {
12            "platformName": "Android",
13            "deviceName": "emulator-5554",
14            "appPackage": "com.code2lead.kwad",
15            "appActivity": ".MainActivity",
16            "automationName": "UiAutomator2"
17        }
18        # Start the Appium client
19        self.driver = webdriver.Remote("http://localhost:4723/wd/hub", desired_caps)
20        self.driver.implicitly_wait(20)
```

The IDE's output window at the bottom shows the execution results for the test `test3.AndroidEnterValueTest`. It indicates that all tests passed: `Tests passed: 1 of 1 test - 31 sec 977 ms`.

```

19     self.driver = webdriver.Remote("http://localhost:4723/wd/hub", desired_caps)
20     self.driver.implicitly_wait(20)
21
22     # Close the Appium client
23     def tearDown(self):
24         self.driver.quit()
25
26         # TC: ID002_userCanEnterValueIntoField
27         # 1. On my page: check if my page is loaded
28         # 2. Click on "Enter some value" field
29         # 3. Enter "hello world" in the "Enter some value" button
30         # 4. Click "Submit" button
31         # 5. Check if values "hello world" are displayed
32
33     def testID003_userCanEnterValueIntoField(self):
34         # 1. On my page: check if my page is loaded
35         self.driver.find_element(AppiumBy.XPATH, "//*[@text='Appium Demo']").is_displayed()
36         # 2. Click on "Enter some value" field
37         el = self.driver.find_element(AppiumBy.ACCESSIBILITY_ID, "Btn1")
38         self.assertEqual(el.is_displayed(), True)

```

AndroidEnterValueTest

for test3.AndroidEnterValueTest

» ✓ Tests passed: 1 of 1 test – 31 sec 977 ms


```
test1.py × test2.py × test3.py ×
37     el = self.driver.find_element(AppiumBy.ACCESSIBILITY_ID, "Btn1")
38     self.assertTrue(el.is_displayed())
39     el.click()
40     # 3. Enter "hello world" in the "Enter some value" button
41     field = self.driver.find_element(AppiumBy.CLASS_NAME, "android.widget.EditText")
42     field.send_keys("hello world!")
43     # 4. Click "Submit" button
44     button = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Btn1")
45     self.assertTrue(button.is_displayed())
46     button.click()
47     # 5. Check if values "hello world" are displayed
48     hello1 = self.driver.find_element(AppiumBy.XPATH, "//*[@resource-id='com.code2lead.kwad:id/Et1']")
49     self.assertTrue(hello1.is_displayed())
50     hello2 = self.driver.find_element(AppiumBy.ID, "com.code2lead.kwad:id/Tv1")
51     self.assertTrue(hello2.is_displayed())
52
53
54     if __name__ == '__main__':
55         unittest.main()
56
AndroidEnterValueTest
test3.AndroidEnterValueTest ×
Tests passed: 1 of 1 test – 31 sec 977 ms
```