

**POSTGRADUATE STUDY „Software Tester”**

**WSB Merito University in Warsaw**

**Mobile App Testing with Appium**



## Table of contents

Project Goal.....	3
Environment.....	3
Test Scenario .....	3
Test Cases.....	4
ID: TC_REGISTRATION_1_0 - Account registration using a blank form .....	4
ID: TC_REGISTRATION_1_1 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number, 1 special character .....	4
ID: TC_REGISTRATION_1_2 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number .....	5
ID: TC_REGISTRATION_1_3 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 special character.....	5
ID: TC_REGISTRATION_1_4 - Account registration using a password consisting of lowercase letters, 1 digit, 1 special character .....	6
ID: TC_REGISTRATION_1_5 - Account registration using a password consisting of uppercase letters, 1 digit, 1 special character .....	6
ID: TC_REGISTRATION_1_6 - Account registration using a too short password .....	7
Automation of Test Cases .....	8
Test Results .....	30
Test Summary .....	30

## Project Goal

Automation of tests to ensure effective and repeatable testing.

Tester's name	Application under test
Anna Pawełek	Headspace

## Environment

Emulator: eSMART

Android: 13

Android Studio Flamingo 2022.2.1

Appium Doctor 1.16.2

Android Debug Bridge 1.0.41

Appium 1.22.3

Appium Inspector 2023.4.3

Node 18.16.0

Npm 9.6.5

Python 3.11.3

Java 19.0.1

PyCharm Community Edition 2023.1

Android Studio Flamingo 2022.2.1

## Test Scenario

Account Registration via Email

Mandatory fields of the registration form:

- First name
- Last name
- E-mail address
- Password

Password requirements:

Consists of 8+ characters and at least 3 of the following:

- 1 lowercase letter
- 1 capital letter
- 1 digit
- 1 special character

## Test Cases

<i>ID</i>	<i>Test Case Name</i>
TC_REGISTRATION_1_0	Account registration using a blank form
TC_REGISTRATION_1_1	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number, 1 special character
TC_REGISTRATION_1_2	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number
TC_REGISTRATION_1_3	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 special
TC_REGISTRATION_1_4	Account registration using a password consisting of lowercase letters, 1 digit, 1 special character
TC_REGISTRATION_1_5	Account registration using a password consisting of uppercase letters, 1 digit, 1 special character
TC_REGISTRATION_1_6	Account registration using a too short password

ID: TC\_REGISTRATION\_1\_0 - Account registration using a blank form

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Leave all the fields blank.
- 4 Click "Create an account".

**Expected result:**

The account is not created.

ID: TC\_REGISTRATION\_1\_1 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number, 1 special character

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter a valid password in the password field.
7. Click "Create an account".

**Expected result:**

The account is created.

ID: TC\_REGISTRATION\_1\_2 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter a valid password in the password field.
7. Click "Create an account".

**Expected result:**

The account is created.

ID: TC\_REGISTRATION\_1\_3 - Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 special character

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter a valid password in the password field.
7. Click "Create an account".

**Expected result:**

The account is created.

ID: TC\_REGISTRATION\_1\_4 - Account registration using a password consisting of lowercase letters, 1 digit, 1 special character

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter a valid password in the password field.
7. Click "Create an account".

**Expected result:**

The account is created.

ID: TC\_REGISTRATION\_1\_5 - Account registration using a password consisting of uppercase letters, 1 digit, 1 special character

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter a valid password in the password field.
7. Click "Create an account".

**Expected result:**

The account is created.

**Preconditions:**

The environment is set, the application is installed on the emulator. The user is not logged in.

**Action:**

- 1 Open the Headspace app on the emulator.
- 2 Click "Continue with email".
- 3 Enter a valid name in first name field.
- 4 Enter a valid last name in last name field.
- 5 Enter a valid email in the email address field.
6. Enter too short password (max 7 characters) in the password field.
7. Click "Create an account".

**Expected result:**

The account is not created.

## Automation of Test Cases

Link to the recording of the test execution: <https://vimeo.com/865577235/5c064b4d87?share=copy>

```
test_tc_registration.py x
1  import unittest
2      from appium import webdriver
3      from appium.webdriver.common.appiumby import AppiumBy
4      from customLogger.custom_logger import customLogger as cl
5
6
7  class RegistrationTest(unittest.TestCase):
8
9      def setUp(self):
10         desired_caps = {}
11         desired_caps['platformName'] = 'Android'
12         desired_caps['platformVersion'] = '13'
13         desired_caps['deviceName'] = 'eSMART'
14         desired_caps['appPackage'] = 'com.getsomeheadspace.android'
15         desired_caps['appActivity'] = '.splash.SplashActivity'
16         desired_caps['automationName'] = 'UiAutomator2'
17         desired_caps['uiautomator2ServerLaunchTimeout'] = 200000
18         desired_caps['uiautomator2ServerInstallTimeout'] = 200000
19         desired_caps['appWaitForLaunch'] = 'false'
20         desired_caps['unicodeKeyboard'] = True
21         desired_caps['resetKeyboard'] = True
22
23         localAppiumUrl = 'http://127.0.0.1:4723/wd/hub'
24
25         # initializing the WebDriver
26         self.driver = webdriver.Remote(localAppiumUrl, desired_caps)
27
28         # creating the object for the Custom Logger class
29         log = cl()
30
31         # saving a comment about the status of the executed step
```



```

32     log.info("Launched the app")
33
34     def tearDown(self):
35         self.driver.quit()
36
37     def test_registration_1_0_blank_fields(self):
38         # LOCATORS for the Welcome to Headspace page
39         continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
40
41         # LOCATORS for the Sign Up page
42         signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
43         isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
44
45         # TEST DATA
46         # all fields are left blank
47
48         # setting the implicitly wait for 35s
49         self.driver.implicitly_wait(35)
50
51         # locating the "Continue with email" button
52         element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
53
54         # clicking the "Continue with email" button
55         element.click()
56
57         # creating the object for the Custom Logger class
58         log = cl()
59
60         # saving a comment about the status of the executed step
61         log.info("Clicked the Continue with email button")
62

```

```

63 # checking if clicking the "Continue with email" button redirected to the Sign Up page
64 element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
65 self.assertTrue(element.is_displayed())
66
67 # taking a screenshot
68 self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
69                             "/TC1_0_SingUpPageConfirmation.png")
70
71 # saving a comment about the status of the executed step
72 log.info("Took the TC_1_0_SingUpPageConfirmation screenshot")
73 log.info("Sing Up page opened")
74
75 # locating the isThePasswordCorrect indicator and checking if the password is correct or not
76 try:
77     element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
78     self.assertFalse(element.is_displayed())
79 except:
80     # saving a comment about the status of the executed step
81     log.info("Password is not correct, an account cannot be created")
82 finally:
83     # taking a screenshot and saving a comment about the status of the executed step
84     log.info("Password is not correct, an account cannot be created")
85     self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
86                                 "/TC1_0_PasswordCheck.png")
87     log.info("Took the TC_1_0_PasswordCheck screenshot")
88
89 def test_registration_1_1_passwordContaining_upperCaseCharacter_lowerCaseCharacters_1number_1specialCharacter(self):
90     # LOCATORS for the Welcome to Headspace page
91     continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
92
93     # LOCATORS for the Sign Up page

```

```

94     signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
95     firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id
96     lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
97     emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
98     password = "com.getsomeheadspace.android:id/passwordEditText" # id
99     isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
100
101     # TEST DATA
102     firstNameValue = "Sheldon"
103     lastNameValue = "Cooper"
104     emailAddressValue = "sc@imaginary.com"
105     passwordValue = "P@sswor1"
106
107     # setting the implicitly wait for 35s
108     self.driver.implicitly_wait(35)
109
110     # locating the "Continue with email" button
111     element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
112
113     # clicking the "Continue with email" button
114     element.click()
115
116     # creating the object for the Custom Logger class
117     log = cl()
118
119     # saving a comment about the status of the executed step
120     log.info("Clicked the Continue with email button")
121
122     # checking if clicking the "Continue with email" button redirected to the Sign Up page
123     element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
124     self.assertTrue(element.is_displayed())

```

```
125
126 # taking a screenshot
127 self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
128                             "/TC1_1_SingUpPageConfirmation.png")
129 # saving a comment about the status of the executed step
130 log.info("Took the TC_1_1_SingUpPageConfirmation screenshot")
131 log.info("Sing Up page opened")
132
133 # locating the first name field
134 element = self.driver.find_element(AppiumBy.ID, firstName)
135
136 # clicking the first name field
137 element.click()
138
139 # saving a comment about the status of the executed step
140 log.info("Clicked the First name field")
141
142 # entering the first name
143 element.send_keys(firstNameValue)
144
145 # saving a comment about the status of the executed step
146 log.info("First name entered")
147
148 # locating the last name field
149 element = self.driver.find_element(AppiumBy.ID, lastName)
150
151 # clicking the last name field
152 element.click()
153
154 # saving a comment about the status of the executed step
155 log.info("Clicked the Last name field")
```

```
156
157     # entering the last name|
158     element.send_keys(lastNameValue)
159
160     # saving a comment about the status of the executed step
161     log.info("Last name entered")
162
163     # locating the email address field
164     element = self.driver.find_element(AppiumBy.ID, emailAddress)
165
166     # clicking the email address field
167     element.click()
168
169     # saving a comment about the status of the executed step
170     log.info("Clicked the Email address field")
171
172     # entering the email address
173     element.send_keys(emailAddressValue)
174
175     # saving a comment about the status of the executed step
176     log.info("Email address entered")
177
178     # locating the password field
179     element = self.driver.find_element(AppiumBy.ID, password)
180
181     # clicking the password field
182     element.click()
183
184     # saving a comment about the status of the executed step
185     log.info("Clicked the Password field")
186
```

```

187         # entering the password
188         element.send_keys(passwordValue)
189
190         # saving a comment about the status of the executed step
191         log.info("Password entered")
192
193         # locating the isThePasswordCorrect indicator and checking if the password is correct or not
194         try:
195             element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
196         except:
197             # taking a screenshot and saving a comment about the status of the executed step
198             log.info("Password is not correct, an account cannot be created")
199             assert False
200         else:
201             self.assertTrue(element.is_displayed())
202             # saving a comment about the status of the executed step
203             log.info("Password is correct, an account can be created")
204         finally:
205             # taking a screenshot and saving a comment about the status of the executed step
206             self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
207                                         "/TC1_1_PasswordCheck.png")
208             log.info("Took the TC_1_1_PasswordCheck screenshot")
209
210
211     def test_registration_1_2_passwordContaining_upperCaseCharacter_lowerCaseCharacters_1number(self):
212         # LOCATORS for the Welcome to Headspace page
213         continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
214
215         # LOCATORS for the Sign Up page
216         signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
217         firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id

```



```

218     lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
219     emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
220     password = "com.getsomeheadspace.android:id/passwordEditText" # id
221     isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
222
223     # TEST DATA
224     emailAddressValue = "hw@blabla.com"
225     firstNameValue = "Howard"
226     lastNameValue = "Wolowitz"
227     passwordValue = "Passwor1"
228
229     # setting the implicitly wait for 35s
230     self.driver.implicitly_wait(35)
231
232     # locating the "Continue with email" button
233     element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
234
235     # clicking the "Continue with email" button
236     element.click()
237
238     # creating the object for the Custom Logger class
239     log = cl()
240
241     # saving a comment about the status of the executed step
242     log.info("Clicked the Continue with email button")
243
244     # checking if clicking the "Continue with email" button redirected to the Sign Up page
245     element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
246     self.assertTrue(element.is_displayed())
247
248     # taking a screenshot

```

```
249 self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
250                               "/TC1_2_SingUpPageConfirmation.png")
251 # saving a comment about the status of the executed step
252 log.info("Took the TC_1_2_SingUpPageConfirmation screenshot")
253 log.info("Sing Up page opened")
254
255 # locating the first name field
256 element = self.driver.find_element(AppiumBy.ID, firstName)
257
258 # clicking the first name field
259 element.click()
260
261 # saving a comment about the status of the executed step
262 log.info("Clicked the First name field")
263
264 # entering the first name
265 element.send_keys(firstNameValue)
266
267 # saving a comment about the status of the executed step
268 log.info("First name entered")
269
270 # locating the last name field
271 element = self.driver.find_element(AppiumBy.ID, lastName)
272
273 # clicking the last name field
274 element.click()
275
276 # saving a comment about the status of the executed step
277 log.info("Clicked the Last name field")
278
279 # entering the last name
```



```
280     element.send_keys(lastNameValue)
281
282     # saving a comment about the status of the executed step
283     log.info("Last name entered")
284
285     # locating the email address field
286     element = self.driver.find_element(AppiumBy.ID, emailAddress)
287
288     # clicking the email address field
289     element.click()
290
291     # saving a comment about the status of the executed step
292     log.info("Clicked the Email address field")
293
294     # entering the email address
295     element.send_keys(emailAddressValue)
296
297     # saving a comment about the status of the executed step
298     log.info("Email address entered")
299
300     # locating the password field
301     element = self.driver.find_element(AppiumBy.ID, password)
302
303     # clicking the password field
304     element.click()
305
306     # saving a comment about the status of the executed step
307     log.info("Clicked the Password field")
308
309     # entering the password
310     element.send_keys(passwordValue)
```

```

311
312     # saving a comment about the status of the executed step
313     log.info("Password entered")
314
315     # locating the isThePasswordCorrect indicator and checking if the password is correct or not
316     try:
317         element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
318     except:
319         # taking a screenshot and saving a comment about the status of the executed step
320         log.info("Password is not correct, an account cannot be created")
321         assert False
322     else:
323         self.assertTrue(element.is_displayed())
324         # saving a comment about the status of the executed step
325         log.info("Password is correct, an account can be created")
326     finally:
327         # taking a screenshot and saving a comment about the status of the executed step
328         self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
329                                     "/TC1_2_PasswordCheck.png")
330         log.info("Took the TC_1_2_PasswordCheck screenshot")
331
332     def test_registration_1_3_passwordContaining_1upperCaseCharacter_lowerCaseCharacters_1specialCharacter(self):
333         # LOCATORS for the Welcome to Headspace page
334         continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
335
336         # LOCATORS for the Sign Up page
337         signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
338         firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id
339         lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
340         emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
341         password = "com.getsomeheadspace.android:id/passwordEditText" # id
342         isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
343
344         # TEST DATA
345         emailAddressValue = "leonard@hofstadter.pl"
346         firstNameValue = "Leonard"
347         lastNameValue = "Hofstadter"
348         passwordValue = "P@ssword"
349
350         # setting the implicitly wait for 35s
351         self.driver.implicitly_wait(35)
352
353         # locating the "Continue with email" button
354         element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
355
356         # clicking the "Continue with email" button
357         element.click()
358
359         # creating the object for the Custom Logger class
360         log = cl()
361
362         # saving a comment about the status of the executed step
363         log.info("Clicked the Continue with email button")
364
365         # checking if clicking the "Continue with email" button redirected to the Sign Up page
366         element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
367         self.assertTrue(element.is_displayed())
368
369         # taking a screenshot
370         self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
371                                     "/TC1_3_SingUpPageConfirmation.png")
372

```

```
373 # saving a comment about the status of the executed step
374 log.info("Took the TC_1_3_SingUpPageConfirmation screenshot")
375 log.info("Sing Up page opened")
376
377 # locating the first name field
378 element = self.driver.find_element(AppiumBy.ID, firstName)
379
380 # clicking the first name field
381 element.click()
382
383 # saving a comment about the status of the executed step
384 log.info("Clicked the First name field")
385
386 # entering the first name
387 element.send_keys(firstNameValue)
388
389 # saving a comment about the status of the executed step
390 log.info("First name entered")
391
392 # locating the last name field
393 element = self.driver.find_element(AppiumBy.ID, lastName)
394
395 # clicking the last name field
396 element.click()
397
398 # saving a comment about the status of the executed step
399 log.info("Clicked the Last name field")
400
401 # entering the last name
402 element.send_keys(lastNameValue)
403
```

```
404         # saving a comment about the status of the executed step
405         log.info("Last name entered")
406
407         # locating the email address field
408         element = self.driver.find_element(AppiumBy.ID, emailAddress)
409
410         # clicking the email address field
411         element.click()
412
413         # saving a comment about the status of the executed step
414         log.info("Clicked the Email address field")
415
416         # entering the email address
417         element.send_keys(emailAddressValue)
418
419         # saving a comment about the status of the executed step
420         log.info("Email address entered")
421
422         # locating the password field
423         element = self.driver.find_element(AppiumBy.ID, password)
424
425         # clicking the password field
426         element.click()
427
428         # saving a comment about the status of the executed step
429         log.info("Clicked the Password field")
430
431         # entering the password
432         element.send_keys(passwordValue)
433
434         # saving a comment about the status of the executed step
```



```

435     log.info("Password entered")
436
437     # locating the isThePasswordCorrect indicator and checking if the password is correct or not
438     try:
439         element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
440     except:
441         # taking a screenshot and saving a comment about the status of the executed step
442         log.info("Password is not correct, an account cannot be created")
443         assert False
444     else:
445         self.assertTrue(element.is_displayed())
446         # saving a comment about the status of the executed step
447         log.info("Password is correct, an account can be created")
448     finally:
449         # taking a screenshot and saving a comment about the status of the executed step
450         self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
451                                     "/TC1_3_PasswordCheck.png")
452         log.info("Took the TC_1_3_PasswordCheck screenshot")
453
454     def test_registration_1_4_passwordContaining_lowerCaseCharacters_1number_1specialCharacter(self):
455
456         # LOCATORS for the Welcome to Headspace page
457         continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
458
459         # LOCATORS for the Sign Up page
460         signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
461         firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id
462         lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
463         emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
464         password = "com.getsomeheadspace.android:id/passwordEditText" # id
465         isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
466
467         # TEST DATA
468         emailAddressValue = "penny@hof.com"
469         firstNameValue = "Penny"
470         lastNameValue = "Hofstadter"
471         passwordValue = "p@sswor1"
472
473         # setting the implicitly wait for 35s
474         self.driver.implicitly_wait(35)
475
476         # locating the "Continue with email" button
477         element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
478
479         # clicking the "Continue with email" button
480         element.click()
481
482         # creating the object for the Custom Logger class
483         log = cl()
484
485         # saving a comment about the status of the executed step
486         log.info("Clicked the Continue with email button")
487
488         # checking if clicking the "Continue with email" button redirected to the Sign Up page
489         element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
490         self.assertTrue(element.is_displayed())
491
492         # taking a screenshot
493         self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
494                                     "/TC1_4_SingUpPageConfirmation.png")
495
496         # saving a comment about the status of the executed step

```

```
497     log.info("Took the TC_1_4_SingUpPageConfirmation screenshot")
498     log.info("Sing Up page opened")
499
500     # locating the first name field
501     element = self.driver.find_element(AppiumBy.ID, firstName)
502
503     # clicking the first name field
504     element.click()
505
506     # saving a comment about the status of the executed step
507     log.info("Clicked the First name field")
508
509     # entering the first name
510     element.send_keys(firstNameValue)
511
512     # saving a comment about the status of the executed step
513     log.info("First name entered")
514
515     # locating the last name field
516     element = self.driver.find_element(AppiumBy.ID, lastName)
517
518     # clicking the last name field
519     element.click()
520
521     # saving a comment about the status of the executed step
522     log.info("Clicked the Last name field")
523
524     # entering the last name
525     element.send_keys(lastNameValue)
526
527     # saving a comment about the status of the executed step
```

```
528     log.info("Last name entered")
529
530     # locating the email address field
531     element = self.driver.find_element(AppiumBy.ID, emailAddress)
532
533     # clicking the email address field
534     element.click()
535
536     # saving a comment about the status of the executed step
537     log.info("Clicked the Email address field")
538
539     # entering the email address
540     element.send_keys(emailAddressValue)
541
542     # saving a comment about the status of the executed step
543     log.info("Email address entered")
544
545     # locating the password field
546     element = self.driver.find_element(AppiumBy.ID, password)
547
548     # clicking the password field
549     element.click()
550
551     # saving a comment about the status of the executed step
552     log.info("Clicked the Password field")
553
554     # entering the password
555     element.send_keys(passwordValue)
556
557     # saving a comment about the status of the executed step
558     log.info("Password entered")
```

```

559
560 # locating the isThePasswordCorrect indicator and checking if the password is correct or not
561 try:
562     element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
563 except:
564     # taking a screenshot and saving a comment about the status of the executed step
565     log.info("Password is not correct, an account cannot be created")
566     assert False
567 else:
568     self.assertTrue(element.is_displayed())
569     # saving a comment about the status of the executed step
570     log.info("Password is correct, an account can be created")
571 finally:
572     # taking a screenshot and saving a comment about the status of the executed step
573     self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
574                                "/TC1_4_PasswordCheck.png")
575     log.info("Took the TC_1_4_PasswordCheck screenshot")
576
577 def test_registration_1_5_passwordContaining_upperCaseCharacters_1number_1specialCharacter(self):
578     # LOCATORS for the Welcome to Headspace page
579     continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
580
581     # LOCATORS for the Sign Up page
582     signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
583     firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id
584     lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
585     emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
586     password = "com.getsomeheadspace.android:id/passwordEditText" # id
587     isThePasswordCorrect = "//android.widget.ImageView[@index='1']" # xpath
588
589     # TEST DATA
590     emailAddressValue = "rajesh@imaginary.in"
591     firstNameValue = "Rajesh"
592     lastNameValue = "Koothrappali"
593     passwordValue = "P@SSWOR1"
594
595     # setting the implicitly wait for 35s
596     self.driver.implicitly_wait(35)
597
598     # locating the "Continue with email" button
599     element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
600
601     # clicking the "Continue with email" button
602     element.click()
603
604     # creating the object for the Custom Logger class
605     log = cl()
606
607     # saving a comment about the status of the executed step
608     log.info("Clicked the Continue with email button")
609
610     # checking if clicking the "Continue with email" button redirected to the Sign Up page
611     element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
612     self.assertTrue(element.is_displayed())
613
614     # taking a screenshot
615     self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
616                                "/TC1_5_SingUpPageConfirmation.png")
617
618     # saving a comment about the status of the executed step
619     log.info("Took the TC_1_5_SingUpPageConfirmation screenshot")
620     log.info("Sing Up page opened")

```



```
621
622     # locating the first name field
623     element = self.driver.find_element(AppiumBy.ID, firstName)
624
625     # clicking the first name field
626     element.click()
627
628     # saving a comment about the status of the executed step
629     log.info("Clicked the First name field")
630
631     # entering the first name
632     element.send_keys(firstNameValue)
633
634     # saving a comment about the status of the executed step
635     log.info("First name entered")
636
637     # locating the last name field
638     element = self.driver.find_element(AppiumBy.ID, lastName)
639
640     # clicking the last name field
641     element.click()
642
643     # saving a comment about the status of the executed step
644     log.info("Clicked the Last name field")
645
646     # entering the last name
647     element.send_keys(lastNameValue)
648
649     # saving a comment about the status of the executed step
650     log.info("Last name entered")
651
```

```
652     # locating the email address field
653     element = self.driver.find_element(AppiumBy.ID, emailAddress)
654
655     # clicking the email address field
656     element.click()
657
658     # saving a comment about the status of the executed step
659     log.info("Clicked the Email address field")
660
661     # entering the email address
662     element.send_keys(emailAddressValue)
663
664     # saving a comment about the status of the executed step
665     log.info("Email address entered")
666
667     # locating the password field
668     element = self.driver.find_element(AppiumBy.ID, password)
669
670     # clicking the password field
671     element.click()
672
673     # saving a comment about the status of the executed step
674     log.info("Clicked the Password field")
675
676     # entering the password
677     element.send_keys(passwordValue)
678
679     # saving a comment about the status of the executed step
680     log.info("Password entered")
681
682     # locating the isThePasswordCorrect indicator and checking if the password is correct or not
```

```

683     try:
684         element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
685     except:
686         # taking a screenshot and saving a comment about the status of the executed step
687         log.info("Password is not correct, an account cannot be created")
688         assert False
689     else:
690         self.assertTrue(element.is_displayed())
691         # saving a comment about the status of the executed step
692         log.info("Password is correct, an account can be created")
693     finally:
694         # taking a screenshot and saving a comment about the status of the executed step
695         self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
696                                     "/TC1_5_PasswordCheck.png")
697         log.info("Took the TC_1_5_PasswordCheck screenshot")
698
699 def test_registration_1_6_too_short_password(self):
700     # LOCATORS for the Welcome to Headspace page
701     continueWithEmail = "com.getsomeheadspace.android:id/continueWithEmailButton" # id
702
703     # LOCATORS for the Sign Up page
704     signUpPageTitle = "com.getsomeheadspace.android:id/titleTextView" # id
705     firstName = "com.getsomeheadspace.android:id/firstNameEditText" # id
706     lastName = "com.getsomeheadspace.android:id/lastNameEditText" # id
707     emailAddress = "com.getsomeheadspace.android:id/emailEditText" # id
708     password = "com.getsomeheadspace.android:id/passwordEditText" # id
709     isThePasswordCorrect = "//android.widget.ImageView[@index='1']" #xpath
710
711     # TEST DATA
712     emailAddressValue = "bmaryann@rost.com"
713     firstNameValue = "Bernadette Mary Ann"
714     lastNameValue = "Rostenkowski"
715     passwordValue = "P@sswo1"
716
717
718     # setting the implicitly wait for 35s
719     self.driver.implicitly_wait(35)
720
721     # locating the "Continue with email" button
722     element = self.driver.find_element(AppiumBy.ID, continueWithEmail)
723
724     # clicking the "Continue with email" button
725     element.click()
726
727     # creating the object for the Custom Logger class
728     log = cl()
729
730     # saving a comment about the status of the executed step
731     log.info("Clicked the Continue with email button")
732
733     # checking if clicking the "Continue with email" button redirected to the Sign Up page
734     element = self.driver.find_element(AppiumBy.ID, signUpPageTitle)
735     self.assertTrue(element.is_displayed())
736
737     # taking a screenshot
738     self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
739                                 "/TC1_6_SingUpPageConfirmation.png")
740
741     # saving a comment about the status of the executed step
742     log.info("Took the TC_1_6_SingUpPageConfirmation screenshot")
743     log.info("Sing Up page opened")
744

```

```
745     # locating the first name field
746     element = self.driver.find_element(AppiumBy.ID, firstName)
747
748     # clicking the first name field
749     element.click()
750
751     # saving a comment about the status of the executed step
752     log.info("Clicked the First name field")
753
754     # entering the first name
755     element.send_keys(firstNameValue)
756
757     # saving a comment about the status of the executed step
758     log.info("First name entered")
759
760     # locating the last name field
761     element = self.driver.find_element(AppiumBy.ID, lastName)
762
763     # clicking the last name field
764     element.click()
765
766     # saving a comment about the status of the executed step
767     log.info("Clicked the Last name field")
768
769     # entering the last name
770     element.send_keys(lastNameValue)
771
772     # saving a comment about the status of the executed step
773     log.info("Last name entered")
774
775     # locating the email address field
```

```

776         element = self.driver.find_element(AppiumBy.ID, emailAddress)
777
778         # clicking the email address field
779         element.click()
780
781         # saving a comment about the status of the executed step
782         log.info("Clicked the Email address field")
783
784         # entering the email address
785         element.send_keys(emailAddressValue)
786
787         # saving a comment about the status of the executed step
788         log.info("Email address entered")
789
790         # locating the password field
791         element = self.driver.find_element(AppiumBy.ID, password)
792
793         # clicking the password field
794         element.click()
795
796         # saving a comment about the status of the executed step
797         log.info("Clicked the Password field")
798
799         # entering the password
800         element.send_keys(passwordValue)
801
802         # saving a comment about the status of the executed step
803         log.info("Password entered")
804
805         # locating the isThePasswordCorrect indicator and checking if the password is correct or not
806         try:
807             element = self.driver.find_element(AppiumBy.XPATH, isThePasswordCorrect)
808             self.assertFalse(element.is_displayed())
809         except:
810             # saving a comment about the status of the executed step
811             log.info("Password is not correct, an account cannot be created")
812         finally:
813             # taking a screenshot and saving a comment about the status of the executed step
814             self.driver.save_screenshot("C:/Users/shade/PycharmProjects/pythonProject/AppiumProject/screenshots"
815                                         "/TC1_6_PasswordCheck.png")
816             log.info("Took the TC_1_6_PasswordCheck screenshot")
817
818     if __name__ == '__main__':
819         unittest.main()

```



## Test Results

ID	TEST CASE	TEST DATA	EXPECTED RESULT	TEST RESULT
TC_REGISTRATION_1_0	Account registration using a blank form	none	The account is not created.	PASS
TC_REGISTRATION_1_1	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number, 1 special character	Sheldon Cooper <a href="mailto:sc@imaginary.com">sc@imaginary.com</a> <a href="#">P@sswor1</a>	The account is created.	PASS
TC_REGISTRATION_1_2	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 number	Howard Wolowitz <a href="mailto:hw@blabla.com">hw@blabla.com</a> Passwor1	The account is created.	PASS
TC_REGISTRATION_1_3	Account registration using a password consisting of 1 uppercase letter, lowercase letters, 1 special character	Leonard Hofstadter <a href="mailto:leonard@hofstadter.pl">leonard@hofstadter.pl</a> <a href="#">P@ssword</a>	The account is created.	PASS
TC_REGISTRATION_1_4	Account registration using a password consisting of lowercase letters, 1 digit, 1 special character	Penny Hofstadter <a href="mailto:penny@hof.com">penny@hof.com</a> <a href="#">p@sswor1</a>	The account is created.	FAIL
TC_REGISTRATION_1_5	Account registration using a password consisting of uppercase letters, 1 digit, 1 special character Created:	Rajesh Koothrappali <a href="mailto:rajesh@imaginary.in">rajesh@imaginary.in</a> <a href="#">P@SSWOR1</a>	The account is created.	FAIL
TC_REGISTRATION_1_6	Account registration using a too short password	Bernadette Mary Ann Rostenkovski <a href="mailto:bmaryann@rost.com">bmaryann@rost.com</a> <a href="#">P@sswo1</a>	The account is not created.	PASS

## Test Summary

Out of 7 test cases, 5 confirmed the expected result, while 2 had results different from the required ones. Test cases TC\_REGISTRATION\_1\_4 and TC\_REGISTRATION\_1\_5 revealed errors in the application. The errors have been reported in Jira and included in the project documentation.

[https://github.com/AniaLaRomana/Tester\\_Portfolio](https://github.com/AniaLaRomana/Tester_Portfolio)