

Poznań, 2021-05-21

**STUDIA PODYPLOMOWE „Tester Automatyzujący w Selenium”**  
**CDV POZNAŃ**

MINI PROJEKT – AUTOMATYZACJA TESTÓW DLA APLIKACJI MOBILNEJ  
ANDROID

Cel projektu: Zautomatyzuj wybrane przypadki testowe dla aplikacji mobilnej systemu Android.

Opracowanie przekazane prowadzącemu powinno zawierać:

- Uzupełnioną sekcję: A - do uzupełnienia przez studentów
- B) KOD: Skrypty testowe (pliki .py) w języku Python
- C) Logi

Sprawdził:

mgr inż. Grzegorz Mazur

Opracował/Opracowali:

Marian SKIBIŃSKI

## SEKCJA A

Środowisko testowe, dane konfiguracyjne (hardware/software):

System operacyjny komputera:  
Ubuntu Linux 20.04.2 LTS, 64-bitowy

Dane o komputerze: Procesor Intel® Core™ i5-6200U CPU @  
2.30GHz × 4, 16GB RAM

Testy z wykorzystaniem **emulatora** oraz **urządzenia fizycznego**

Emulator: Pixel 2 5.0 xxhdpi  
Wersja mobilnego system operacyjnego na emulatorze:  
Android 9.0 Pie x86

Urządzenie fizyczne: SAMSUNG Galaxy A5 (2017) SM-A520F  
Wersja mobilnego system operacyjnego na urządzeniu: Android  
8.0.0 Oreo dla ARM

java	Android Studio	node.js	npm	appium
Openjdk version "1.8.0_282"  OpenJDK Runtime Environment build 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08  OpenJDK 64-Bit Server VM build 25.282-b08, mixed mode	4.1.2	v14.16.0.	6.14.11	1.20.2

appium-doctor	Pycharm	python	pip	Appium Python Client	selenium
v.1.16.0	2021.1 (Community Edition), built on April 6, 2021	3.8.5	21.0.1	1.1.0	3.141.0

## Test cases

### ***Detailed information for each test case***

Name of the application under test: **Literowiec**

**Number and name of the test cases** (the same numbers are placed in the source code as comments before each test method; the names below are the same as the names of their respective test methods in the source code):

*Test case No 1 verifies fundamental application functionality and as such should be performed several times to increase probability of finding defects. Use ddt to run it more than once. Before every second run switch letters on the screen to uppercase.*

#### **1. test\_build\_the\_word**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Every second run switch the letters to uppercase by using button with ----- sign on it.
4. From letters scattered on the screen build the word that properly describes the picture. The word should be built inside the red box.

*Expected results:*

The word properly describing the picture being shown on is built.

The button with green arrow appears.

## **2. test\_build\_the\_word\_incorrectly**

*Steps:*

1. Start application
2. Dismiss splash screen
3. From letters scattered on the screen build the word that **incorrectly** describes the picture. The word should be built inside the red box.

*Expected results:*

All letters are placed in the red box, but they do not form the word properly describing the picture. The button with green arrow does NOT appear.

## **3. test\_clicking\_At\_button**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Remember/note the word under the picture, the positions of each labels with letter and the set of letters they contain (make a screenshot)
4. Press the button with @ sign

*Expected results:*

The word under the picture remains unchanged.

Labels with letters should appear in positions different than the ones before clicking the @ button. The set of letters they contain should be the same as the previous one.

#### **4. test\_from\_lowercase\_to\_uppercase**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Click on the button with long ----- sign

*Expected results:*

The word under the picture and all the labels should be changed to uppercase

#### **5. test\_from\_lowercase\_to\_uppercase\_and\_back**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Click on the button with long ----- sign. The word and labels change to uppercase
4. Click on the button with long | sign (this is the same button as in 3., but its label has changed)

*Expected results:*

The word under the picture and all the labels should be in the same registry as they were at the beginning of the test (generally in lowercase, but it is not always the case, eg. Mikołaj → MIKOŁAJ → Mikołaj)

## **6. test\_number\_of\_labels\_is\_correct**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Count the number of letters in the word and the number of scattered labels

*Expected results:*

Both numbers should be the same

## **7. switching\_to\_settings**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Long press on the picture

*Expected results:*

New screen should appear. The new screen fully covers the old one.

The new screen should contain checkable elements.

## **8. test\_switching\_off\_word\_and\_picture**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Go to settings by long touching on the picture
4. Uncheck the field „Nazwa pod obrazkiem”
5. Switch on the radio button „bez Obrazków”
6. Return to MainActivity by pressing back button on the device

*Expected results:*

The picture and the word underneath should not be present/visible.

## **9. test\_switching\_to\_info\_activity**

*Steps:*

1. Start application
2. Dismiss splash screen
3. Go to settings by long touching on the picture
4. Scroll down till *Info* button is seen
5. Click on the *Info* button

*Expected results:*

New screen should appear. The new screen fully covers the old one. The new screen has „Informacje o aplikacji” text in the action bar.

## SEKCJA B

KOD: Skrypty testowe (pliki .py) w języku Python

Kod do pobrania/sklonowania z:

[https://github.com/mskib77/Literowiec\\_POM](https://github.com/mskib77/Literowiec_POM)

Testy znajdują się w następujących modułach:

- tests\_on\_main\_activity.py
- tests\_on\_settings\_page.py

Plik z testowaną aplikacją nosi nazwę *Literowiec.apk* i znajduje się w katalogu **tests** projektu. Aplikację należy zainstalować na emulatorze lub urządzeniu fizycznym.

Zalecany sposób uruchomienia testów: ***python3 suite\_report.py***

Logi z wykonania testów umieszczone zostaną w katalogu **test\_results** w domyślnym katalogu projektu jako pliki html.

## SEKCJA C

LOGI serwera Appium:

Logi przekazano w formie spakowanego archiwum jako załącznik do maila.

Nazwa załączonego archiwum z logami: *2\_Appium\_Logs.7z*

Archiwum zawiera 9 plików z logami serwera Appium z poszczególnych testów. Nazwy plików odpowiadają przypadkom testowym i nazwom funkcji testujących. Numery poprzedzające nazwy plików mają za zadanie ułatwić identyfikację przypadku testowego zarówno w kodzie jak i w dokumentacji.