

Wrocław, 08.04.2018

**STUDIA PODYPŁOMOWE - TESTER OPROGRAMOWANIA**

**Wyższa Szkoła Bankowa Wrocław**

**MINI PROJEKT**

**AUTOMATYZACJA TESTÓW DLA APLIKACJI MOBILNEJ ANDROID**

Cel projektu:

Zautomatyzuj wybrane przypadki testowe dla aplikacji mobilnej systemu Android.

Sprawdził:

mgr inż. Grzegorz Mazur

Opracowały:

Agnieszka Rytwińska

Katarzyna Sycz

## I. Środowisko testowe, dane konfiguracyjne (hardware/software):

- PC: Windows 7 Professional SP1 64-bit Intel Core i3-3240 CPU @ 3.40 GHz 3.40 GHz, RAM 8 GB
- Java 1.8.0\_161
- Android Studio 2.3.1
- Node.js 5.6.0
- npm 5.6.0
- Appium 1.6.4
- Appium-doctor 1.4.3
- PyCharm 2017.1.2
- Python 2.7.13
- Appium Python Client
- Selenium 3.3.1
- Mobile device: Gigaset GS170 with Android 7.0

## II. Wybrane przypadki testowe:

Skrypt: **test\_instalacji\_aplikacji.py**

01. Sprawdzenie, czy aplikacja ApiDemos-debug jest zainstalowana.
02. Sprawdzenie, czy aplikacja ApiDemos-debug nie jest zainstalowana.

Skrypt: **powiadomienia.py**

01. Sprawdzenie, czy powiadomienia się pojawiają.
02. Sprawdzenie, czy powiadomienie „USB debugging connected. Tap to disable USB debugging” znajduje się wśród wyświetlonych powiadomień.
03. Sprawdzenie, czy powiadomienie „USB for file transfer. Tap for more options” znajduje się wśród wyświetlonych powiadomień.

Skrypt: **contact\_manager.py**

01. Sprawdzenie, czy można dodać nowy kontakt.

### III. Informacje szczegółowe dla każdego z przypadków testowych

Skrypt: **test\_instalacji\_aplikacji.py**

Nazwa testowanej aplikacji: **ApiDemos-debug**

Numer i nazwa przypadku testowego:

#### **TC01. Sprawdzenie, czy aplikacja ApiDemos-debug jest zainstalowana.**

Warunki wstępne:

- Aplikacja ApiDemos-debug jest zainstalowana.
- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Uruchom aplikację ApiDemos-debug poprzez uruchomienie skryptu `test_instalacji_aplikacji.py`.

Oczekiwane rezultaty:

Środowisko zwraca wynik OK, jeśli aplikacja jest zainstalowana.

#### **TC02. Sprawdzenie, czy aplikacja ApiDemos-debug nie jest zainstalowana.**

Warunki wstępne:

- Aplikacja ApiDemos-debug jest zainstalowana.
- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Spróbuj uruchomić aplikację ApiDemos-debug poprzez uruchomienie skryptu `test_instalacji_aplikacji.py`.

Oczekiwane rezultaty:

Środowisko zwraca wynik OK, jeśli aplikacja nie jest zainstalowana.

Skrypt **powiadomienia.py**

Nazwa testowanej aplikacji: **ApiDemos-debug**

Numer i nazwa przypadku testowego:

**TC01. Sprawdzenie, czy powiadomienia się pojawiają.**

Warunki wstępne:

- Aplikacja ApiDemos-debug jest zainstalowana.
- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Uruchom aplikację ApiDemos-debug poprzez uruchomienie skryptu powiadomienia.py.

Oczekiwane rezultaty:

Środowisko zwraca wynik OK, jeśli pasek powiadomień się wysuwa i powiadomienia są widoczne.

**TC02. Sprawdzenie, czy powiadomienie USB debugging connected. Tap to disable USB debugging znajduje się wśród wyświetlonych powiadomień.**

Warunki wstępne:

- Aplikacja ApiDemos-debug jest zainstalowana.
- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Otwórz pasek powiadomień aplikacji ApiDemos-debug poprzez uruchomienie skryptu powiadomienia.py.
2. Znajdź następujące powiadomienie wśród wszystkich powiadomień:  
Powiadomienie: USB debugging connected. Tap to disable USB debugging.

Oczekiwane rezultaty:

Jeśli dane powiadomienie zostanie znalezione to test zwraca OK. Jeśli nie to zwraca komunikat FAILED.

**TC03. Sprawdzenie, czy powiadomienie : USB for file transfer. Tap for more options znajduje się wśród wyświetlonych powiadomień.**

Warunki wstępne:

- Aplikacja ApiDemos-debug jest zainstalowana.

- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Zainstaluj i skonfiguruj API i emulator.
2. Otwórz pasek powiadomień.
3. Wykonaj testy instalacji aplikacji ApiDemos-debug poprzez uruchomienie skryptu powiadomienia.py. Znajdź następujące powiadomienie wśród wszystkich powiadomień: powiadomienie.py: USB for file transfer. Tap for more options.

Oczekiwane rezultaty:

Jeśli dane powiadomienie zostanie znalezione to test zwraca OK. Jeśli nie to zwraca komunikat FAILED.

Skrypt: **contact\_manager.py**

Nazwa testowanej aplikacji: **ContactManager**

Numer i nazwa przypadku testowego:

**TC01. Sprawdzenie, czy można dodać kontakt.**

Warunki wstępne:

- Aplikacja **ContactManager** jest zainstalowana.
- Appium i telefon lub emulator są skonfigurowane.

Kroki:

1. Uruchom aplikację **ContactManager**.
2. Naciśnij opcję **Add Contact**.
3. Wpisz imię w polu **Contact Name**.
4. Wpisz numer telefonu w polu **Contact Phone**.
5. Wpisz adres e-mail w polu **Contact Email**.
6. Naciśnij przycisk **Save**.

Oczekiwane rezultaty:

Środowisko zwraca wynik OK, jeśli pola zostały poprawnie wypełnione.