**Aplicatie Android**

**Tehnologii Wireless si Dispozitive Mobile**

Student: Fagadar Ionela-Catalina

Profesor: Iancu Bogdan

**Cuprins**

[**Principalele obiective** 3](#_Toc71543443)

[**Pasi pentru instalare** 3](#_Toc71543444)

[**Partea de mobile (Android)** 3](#_Toc71543445)

[**Partea de server (Python)** 4](#_Toc71543446)

[**Interfata mobila** 5](#_Toc71543447)

[**Referinte** 7](#_Toc71543448)

# **Principalele obiective**

Aplicatia se deschide cu un meniu de alegere a doua optiuni: “Network interfaces” care foloseste serviciul de wifi si returneaza retelele disponibile din jur si “Process image”, unde, am ales sa implementez comunicarea unei aplicatii Android cu un server Python, folosind servicii web REST. Aceste doua aplicatii vor comunica prin intermediul protocolului HTTP, serviciul web de tip RESTful fiind potrivit pentru aceste cazuri de baza. Astfel, aplicatia Android va rula pe un telefon ce va avea acces la internet, aceasta conectandu-se la serverul local realizat cu ajutorul limbajului Python si a framework-ului Flask. Aplicatia va permite selectarea unei poze (din galerie sau o poza implicita) ce va fi trimisa utilizand servicii web REST catre serverul Python, unde se va realiza procesarea imaginii trimise, raspunsul constand in imaginea procesata. Aplicatia va afisa aceasta imagine gata procesata, impreuna cu imaginea sursa.

Aplicatia client (Android) va contine un buton de alegere a imaginii implicte, si un buton de alegere a unei poze din galerie. Butonul pentru procesare va initializa o conexiune cu serverul Python si va trimite, utilizand metoda de tip POST, o imagine catre acesta, urmand ca serverul sa o proceseze (procesare din RGB in grayscale) si sa trimita poza ca si raspuns.

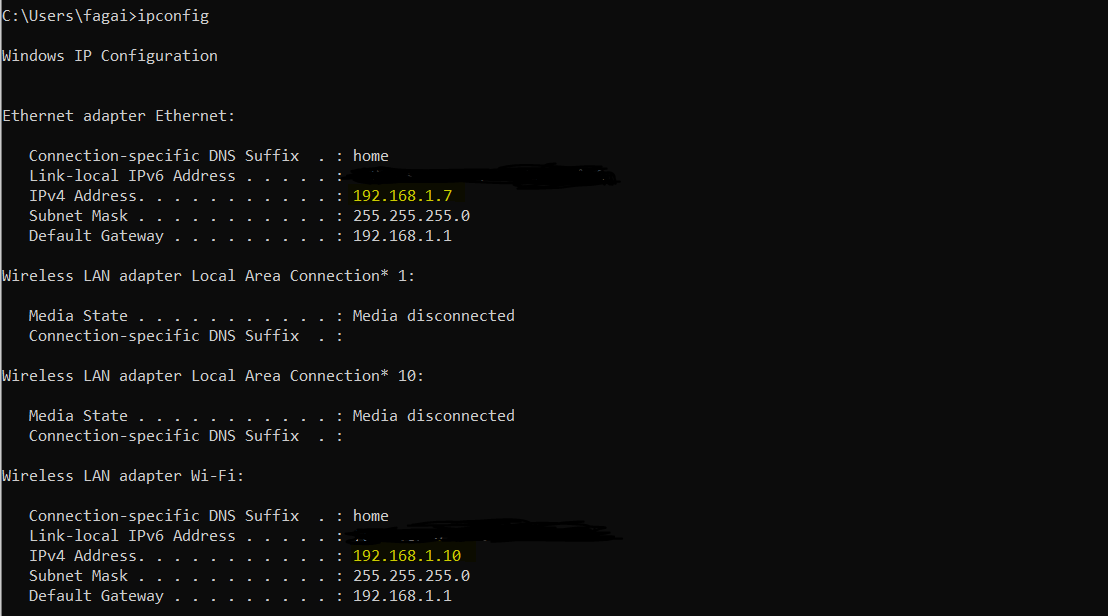
# **Pasi pentru instalare**

## **Partea de mobile (Android)**

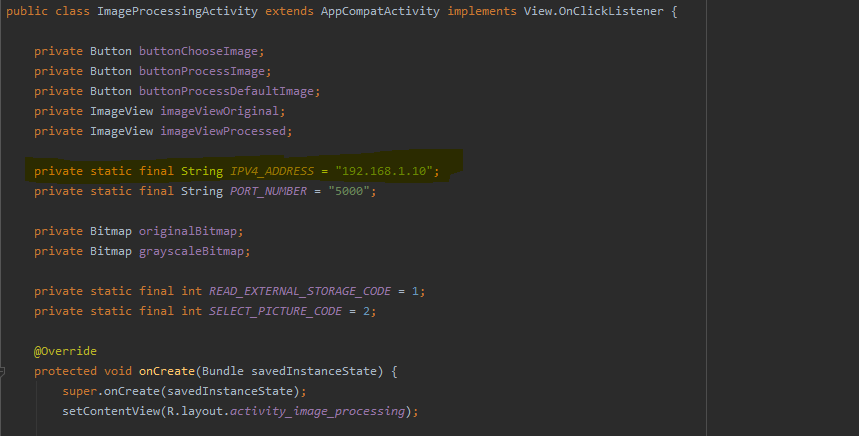
Pentru a rula aceasta aplicatie, sunt necesare:

* Android Studio + Emulator/Telefon cu sistemul de operare Android (https://developer.android.com/studio/)
* Python cu librariile necesare instalate (pil, flask, torch etc.)

Ca prim pas, aplicatia trebuie configurata pentru a comunica cu serverul de python aflat pe calculatorul personal. Pentru aceasta, se deschide linia de comanda si se ruleaza comanda “ipconfig”, pentru a afla adresa de ipv4 personala. Aceasta va inlocui adresa de ipv4 scrisa in aplicatia android, aflata in clasa ImageProcessingActivity, in variabila IPV4\_ADDRESS (linia 46).



Figură - comanda "ipconfig"



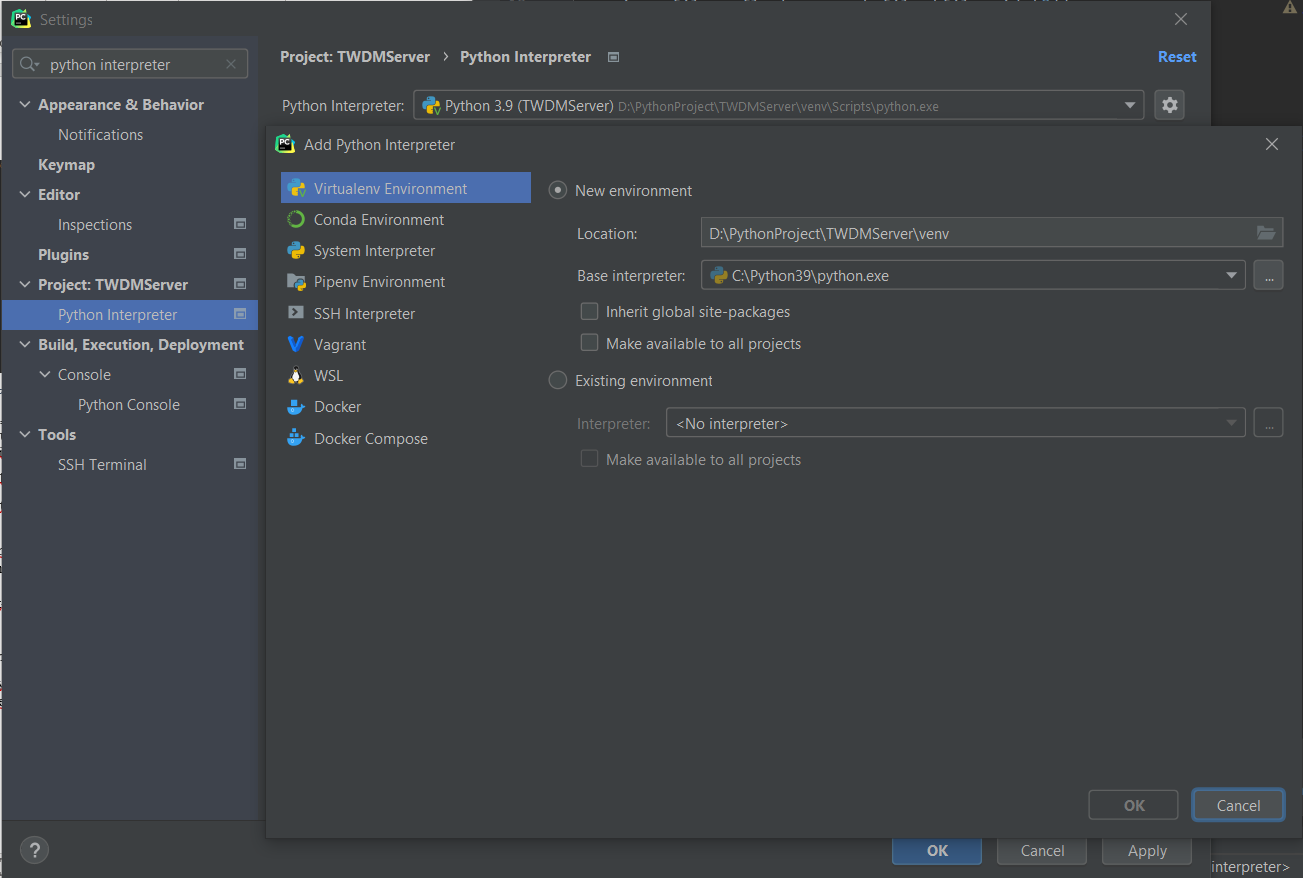
Figură - Portiune cod de schimbat

Aplicatia poate fi utilizata atat pe emulator, cat si pe telefonul personal, ce trebuie sa aiba versiunea e android sdk minim 23. Pentru emulator, este necesar un telefon cu API 27, recomandat fiind pixel 4 API 27 cu play store, aplicatia fiind testata pe acest tip de emulator.

## **Partea de server (Python)**

Pentru partea de server, trebuie instalat python (https://www.python.org/downloads/release/python-395/).

In pycharm, se alege file > settings, apoi python interpreter. Aici se alege un interpretor nou (click pe rotita, add si click virtualenv environment, apoi se alege un nou environment)

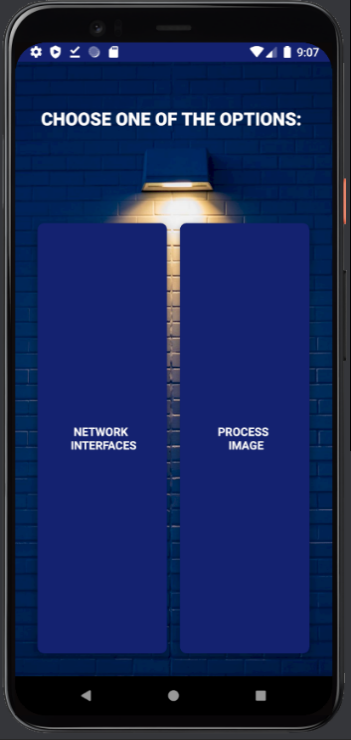


Se instaleaza urmatoarele librarii, folosind terminalul:

* Pil, Flask, Torch folosind comanda:
  + (venv) D:\PythonProject\TWDMServer>**pip install -r requirements.txt**
* Se ruleaza serverul

# **Interfata mobila**

Aplicatia se deschide cu un meniu de alegere, ca in urmatoarea imagine:



Figură - MenuActivity

Alegand optiunea “Network interfaces”, se deschide o pagina ce initial, nu va contine nimic, decat un buton de scanare. Odata cu apasarea butonul de scanare, pe ecran vor fi disponibile retelele wireless:



Figură - WifiActivity

Deoarece este folosit emulatorul, pe ecran va fi disponibila doar o retea wifi, numita AndroidWifi. Acest emulator nu poate accesa retelele wireless locale.

Optiunea de “Process image” va contine 3 butoane: unul pentru a selecta o imagine din galeria proprie, iar altul, pentru a selecta o imagine standard. Dupa ce imaginea este selectata si afisata, se poate apasa butonul “Process” pentru a comunica cu serverul de python si astfel, pentru a procesa imaginea din color in grayscale. Imaginea procesata va fi afisata sub butonul de “Process”.



Figură - ImageProcessingActivity

# **Referinte**

* <https://flask.palletsprojects.com/en/1.1.x/>
* <https://developer.android.com/>
* https://heartbeat.fritz.ai/uploading-images-from-android-to-a-python-based-flask-server-691e4092a95e