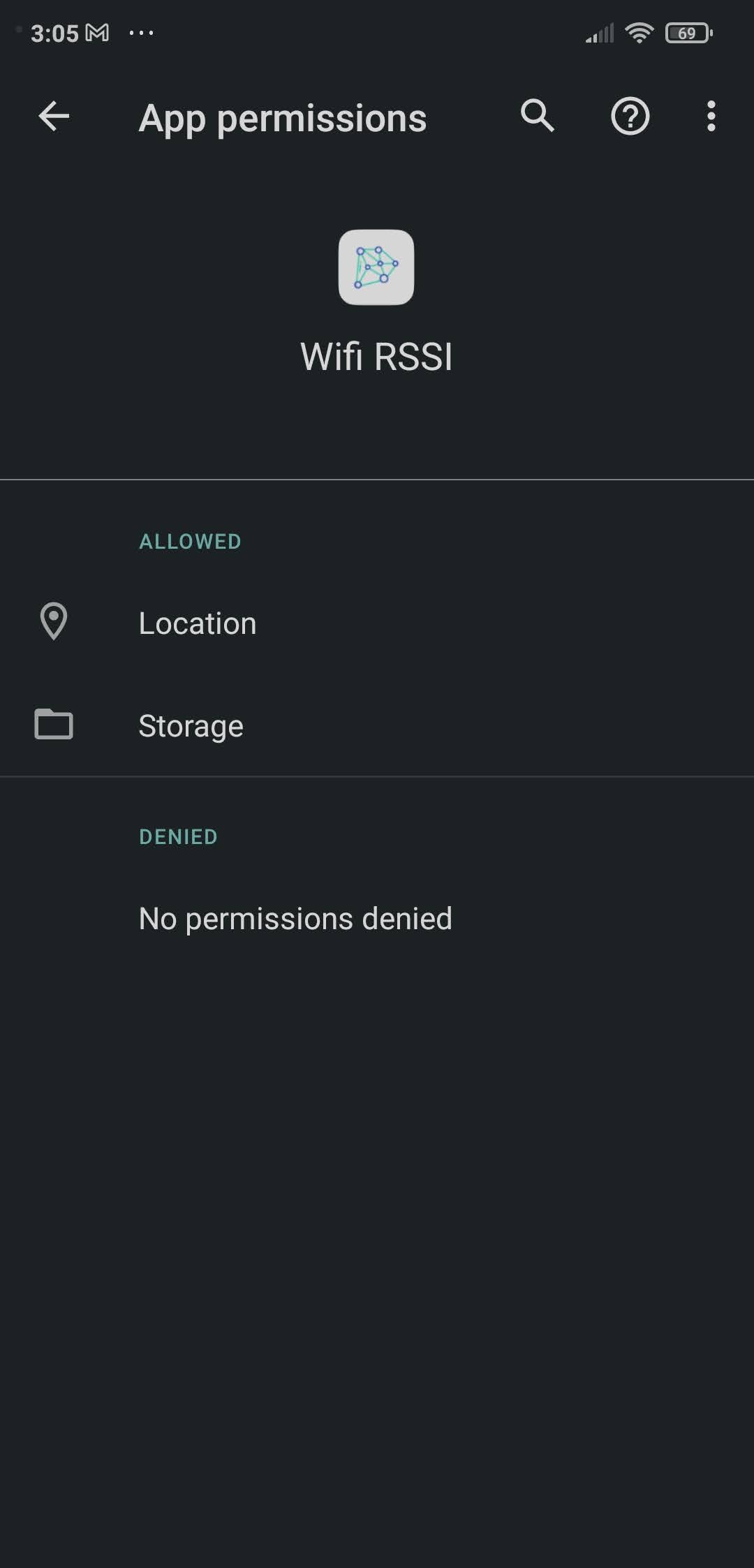
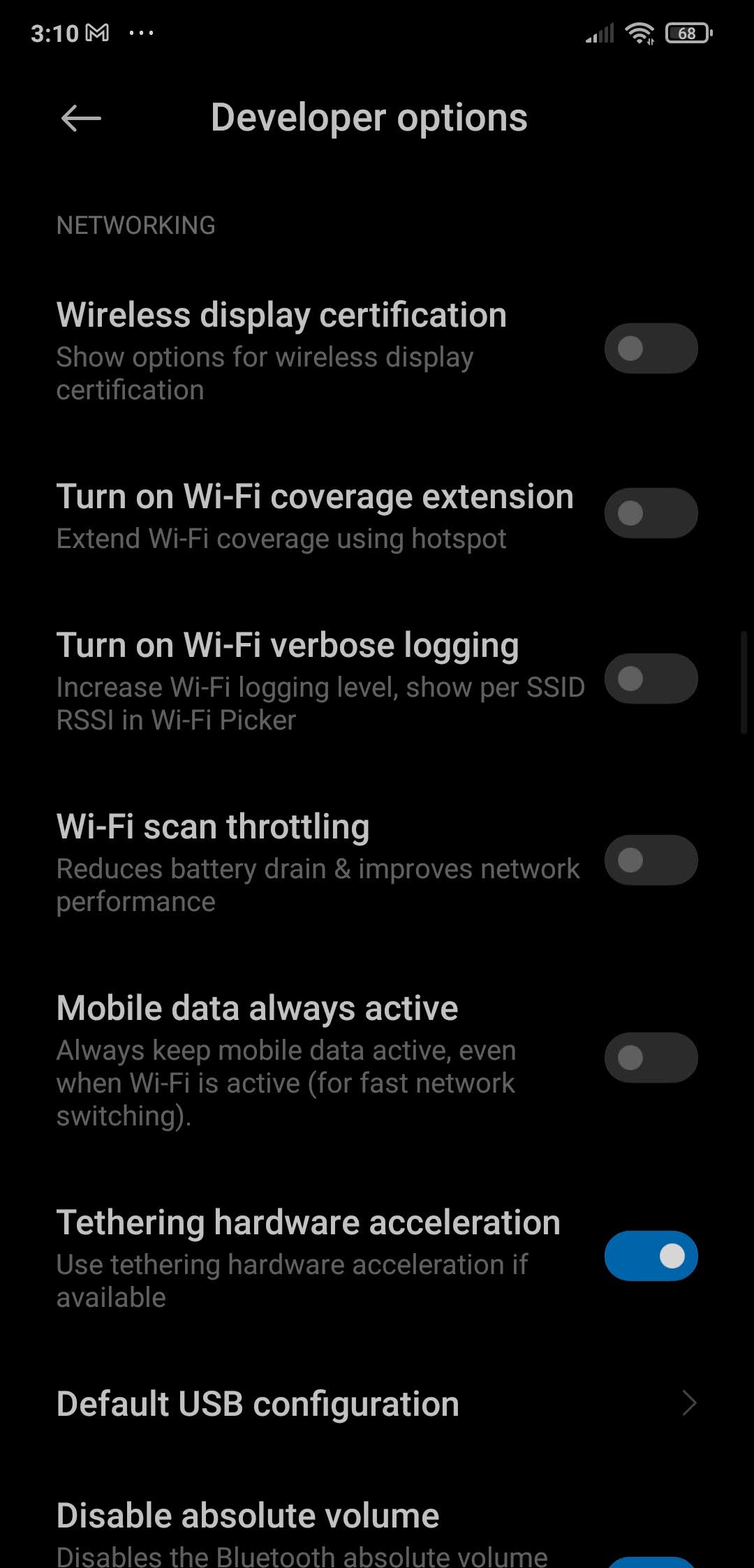
(Give All Required Permissions to App)

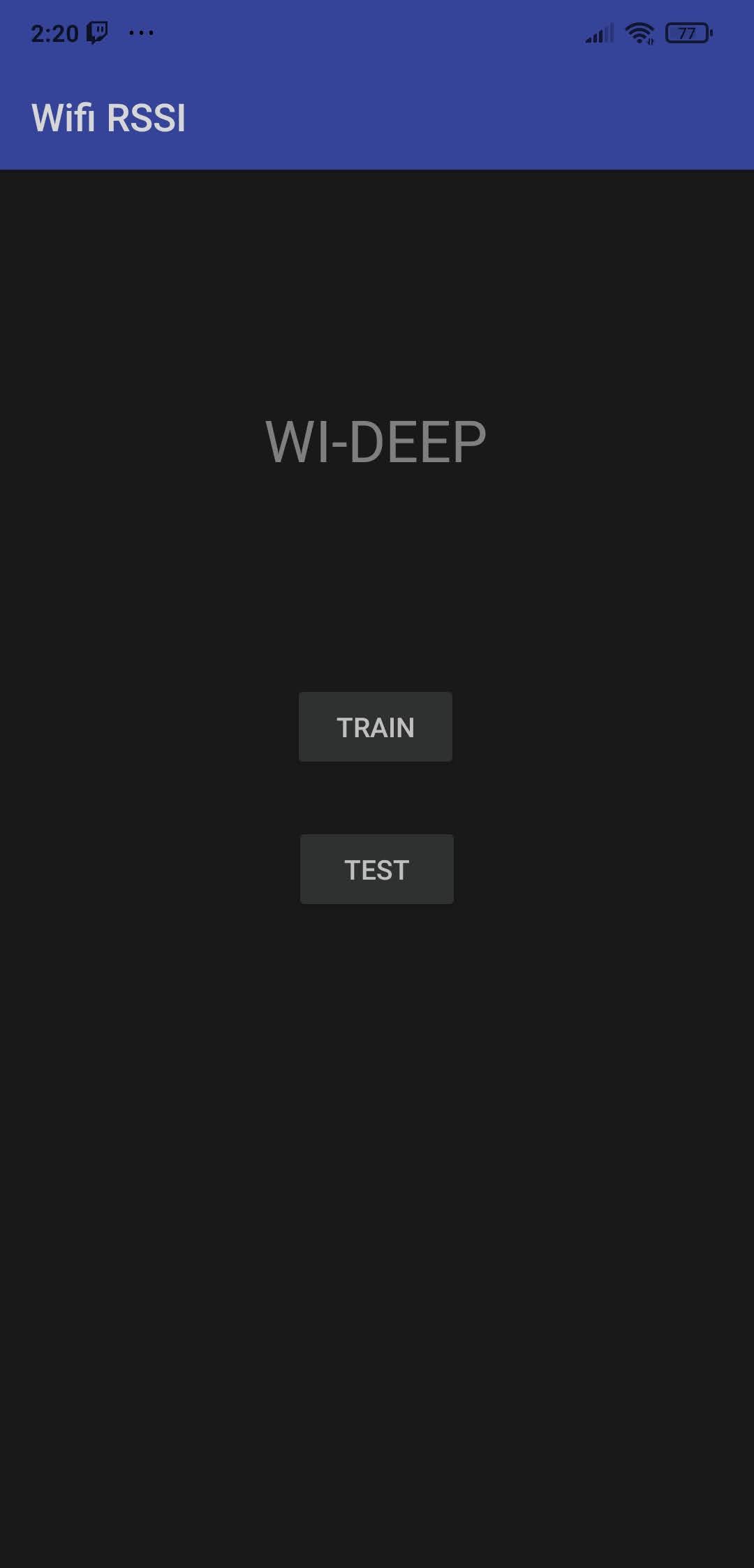


Collecting Data

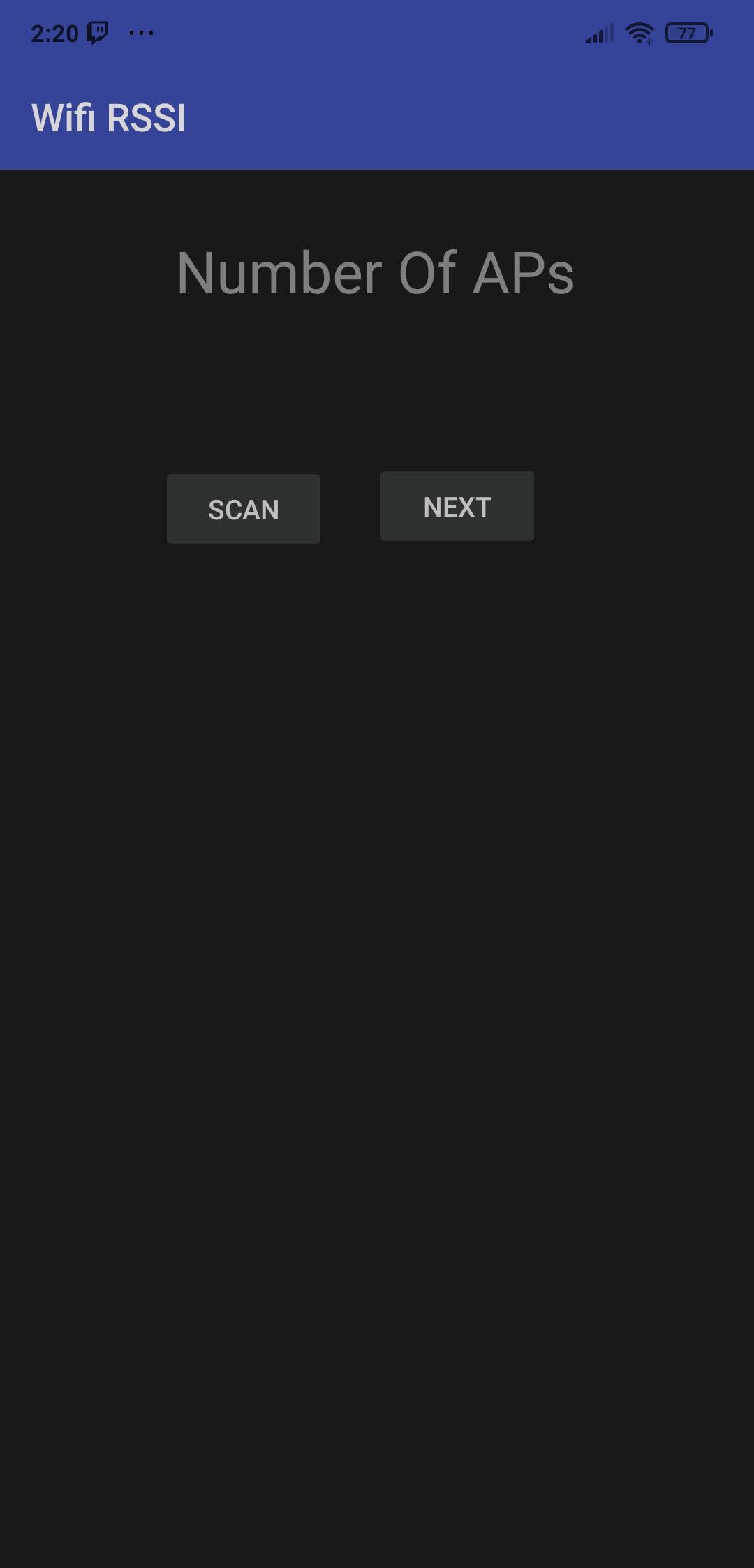
- Disable Wifi Scan Throttling Setting in Android System

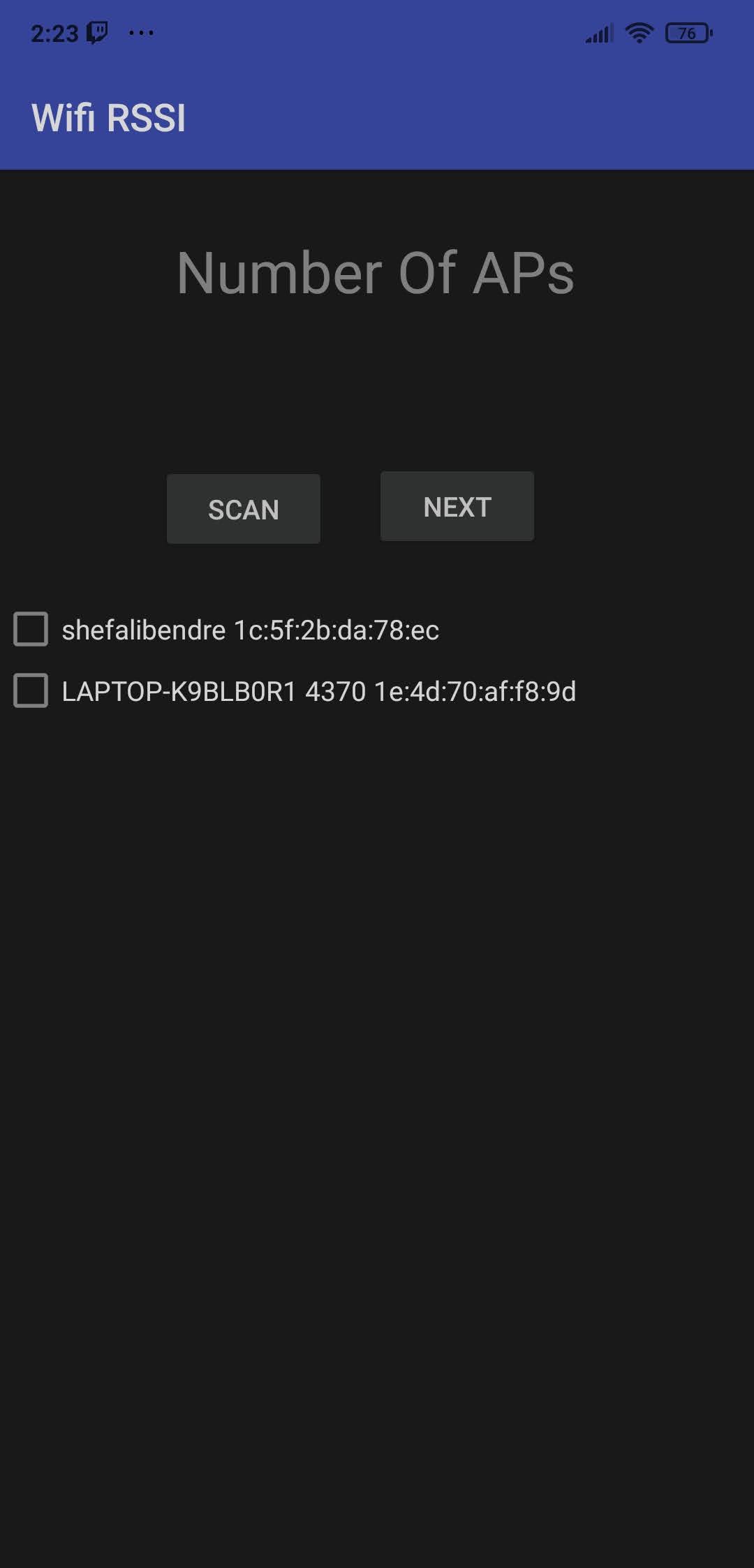


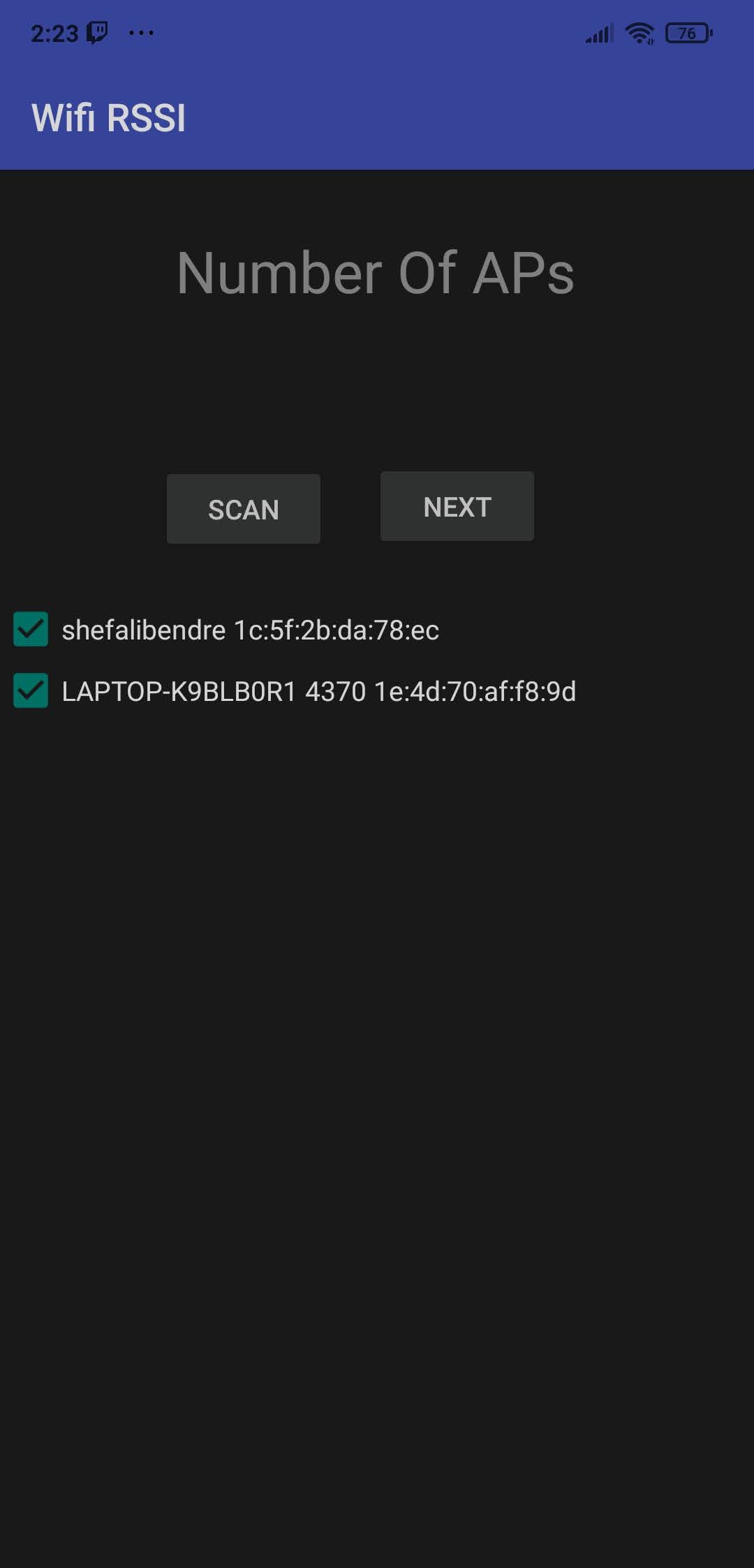
- Open Android App and select Train



- Select Scan and then select From the Available Aps (Make sure that the aps selected for scanning are stable throughout the scanning process)



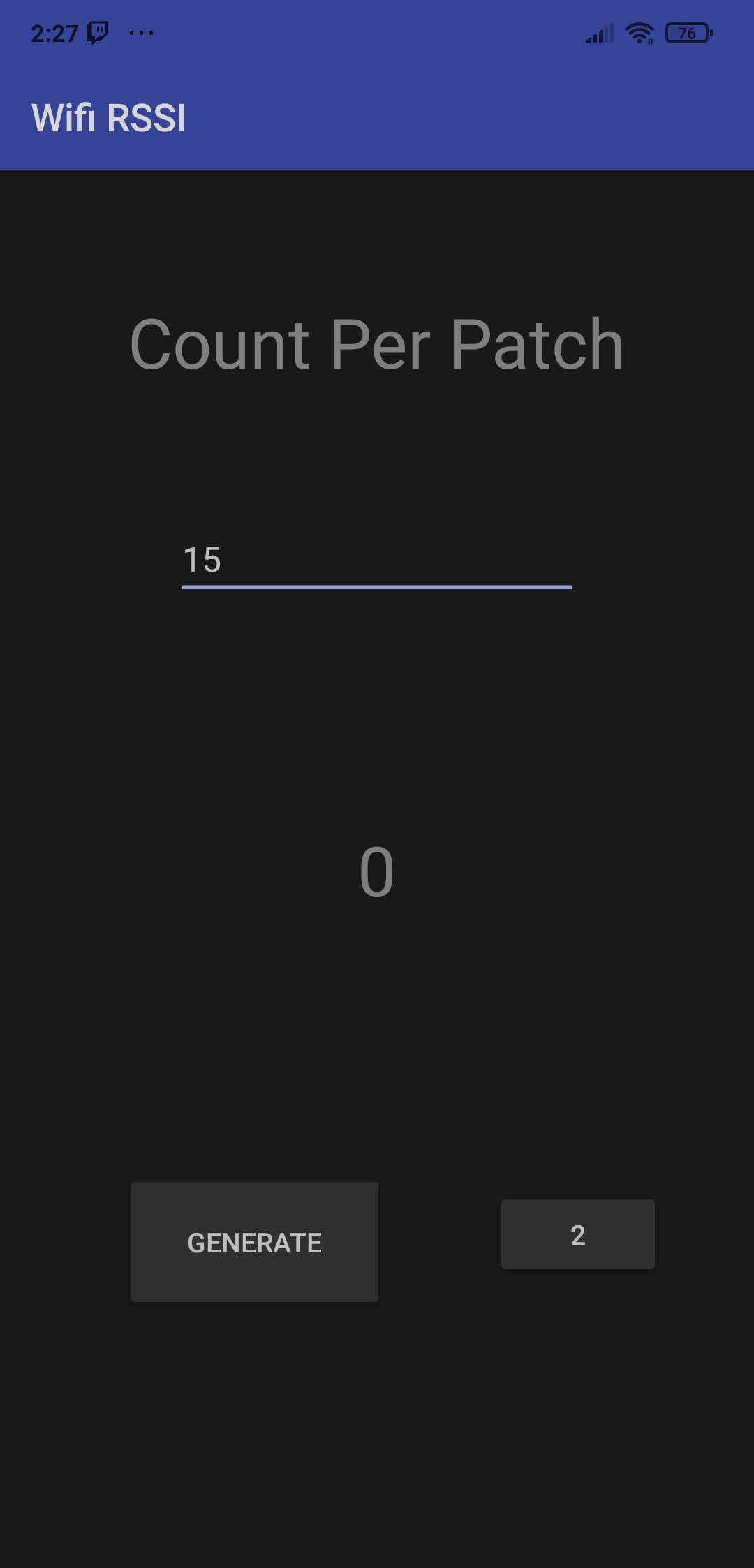




- Collect Fingerprints per Patch and click on generate



* The patch number can be incremented by clicking on the patch number



- A Samples.txt will be created in Internal Storage



Train Data (Aps by Default are 4. Make Appropriate Changes)

- Copy Collected Data in a csv file

- Remove SSID, MAC Address from csv file

- Train model using ann\_train.py

(Changes to be made Before Training

X = dataset.iloc[:,0:4].values

Y = dataset.iloc[:,4].values

Change value of 4 to number of aps

classifier.add(tf.keras.layers.Dense(units = 25, kernel\_initializer= 'uniform', activation = 'relu', input\_dim = 4))

classifier.add(tf.keras.layers.Dense(units = 25, kernel\_initializer= 'uniform', activation = 'relu'))

classifier.add(tf.keras.layers.Dense(units = 5, kernel\_initializer= 'uniform', activation = 'softmax'))

Change input\_dim to number of aps

Add more layers if necessary

Change units in last layer to aps + 1

)

- Model and Transformation will be exported

- Train model using autoencoder\_train.py

(Changes to be made before training

nb\_aps = 4

Change nb\_aps to number of aps

self.fc1 = nn.Linear(nb\_aps, 2)

self.fc2 = nn.Linear(2, 1)

self.fc3 = nn.Linear(1, 2)

self.fc4 = nn.Linear(2, nb\_aps)

Here we can more layers and vary number of nodes

In this case, nb\_aps/2 i.e, 2 and nb\_aps/4 i.e, 1 has been considered

mean\_corrector = nb\_aps/float(4.0 + 1e-10)

Change 4.0 to number of aps with decimal point

)

- Model will be exported

- Create a Dictionary file which will store MAC addresses and its index according to csv file and store it as 'mac\_dictions'

For eg.

Key Type Value

1c:5f:2b:da:78:ec str 0

1e:4d:70:af:f8:9d str 3

1e:96:e6:3d:e2:df str 2

a6:ae:12:0e:37:ff str 1

Deployment

- Install xampp

- Create Wideep\_Project Folder in htdocs and copy Server Files in it

- Change Server IP in Android Project

- Run xampp

- Run rssi\_to\_point.py on Server

- Run App to See Respective Positions