In [3]:

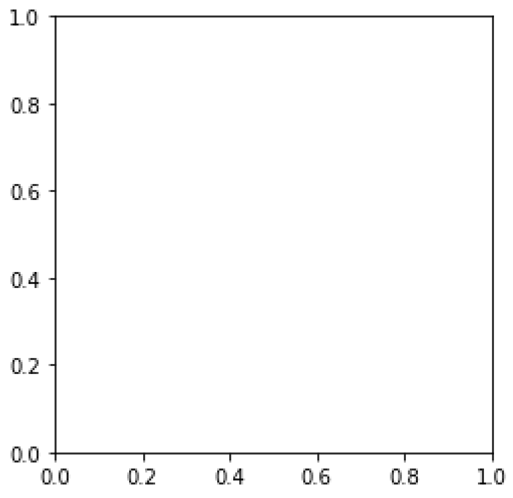
```
"""importing required package"""
from pyzbar import pyzbar         #use to decode the qr codes
import cv2                        #use for the manipulation of the image,videos
import matplotlib.pyplot as plt  #use to plot images
```

In [8]:

```
"""For scanning the QR Code in image"""
# load the input image
image = cv2.imread("/home/shreeagt/Desktop/IP internship/pic3.png")
# find the barcodes in the image and decode each of the barcodes
plt.imshow(image)
plt.show()
barcodes = pyzbar.decode(image)

# loop over the detected barcodes
for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the
    # bounding box surrounding the barcode on the image

#     barcodeType="notfound"
    (x, y, w, h) = barcode.rect
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on
    # our output image we need to convert it to a string first
    barcodeData = barcode.data.decode("utf-8")
    barcodeType = barcode.type
    # draw the barcode data and barcode type on the image
    text = "{} ({})".format(barcodeData, barcodeType)
    cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,0.5, (0, 0, 255)
    # print the barcode type and data to the terminal
    print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
    print(x,y,x+w,y+h)
    # show the output image
plt.imshow(image)
plt.show()
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
 last)
<ipython-input-8-bc3459e24041> in <module>
      2 image = cv2.imread("/home/shreeagt/Desktop/IP internship/pic3.
png")
      3 # find the barcodes in the image and decode each of the barcod
es
----> 4 plt.imshow(image)
      5 plt.show()
      6 barcodes = pyzbar.decode(image)

~/anaconda3/lib/python3.7/site-packages/matplotlib/pyplot.py in imshow
(X, cmap, norm, aspect, interpolation, alpha, vmin, vmax, origin, exte
nt, shape, filternorm, filterrad, imlim, resample, url, data, **kwarg
s)
   2682         filternorm=filternorm, filterrad=filterrad, imlim=imli
m,
   2683         resample=resample, url=url, **({"data": data} if data
 is not
-> 2684         None else {}), **kwargs)
   2685     sci(__ret)
   2686     return __ret

~/anaconda3/lib/python3.7/site-packages/matplotlib/__init__.py in inne
r(ax, data, *args, **kwargs)
   1597     def inner(ax, *args, data=None, **kwargs):
   1598         if data is None:
```

```
-> 1599             return func(ax, *map(sanitize_sequence, args), **k
wargs)
   1600 
   1601         bound = new_sig.bind(ax, *args, **kwargs)

~/anaconda3/lib/python3.7/site-packages/matplotlib/cbook/deprecation.p
y in wrapper(*args, **kwargs)
    367                 f"%(removal)s.  If any parameter follows {nam
e!r}, they "
    368                 f"should be pass as keyword, not positionall
y.")
--> 369         return func(*args, **kwargs)
    370 
    371     return wrapper

~/anaconda3/lib/python3.7/site-packages/matplotlib/cbook/deprecation.p
y in wrapper(*args, **kwargs)
    367                 f"%(removal)s.  If any parameter follows {nam
e!r}, they "
    368                 f"should be pass as keyword, not positionall
y.")
--> 369         return func(*args, **kwargs)
    370 
    371     return wrapper

~/anaconda3/lib/python3.7/site-packages/matplotlib/axes/_axes.py in im
show(self, X, cmap, norm, aspect, interpolation, alpha, vmin, vmax, or
igin, extent, shape, filternorm, filterrad, imlim, resample, url, **kw
args)
   5677                               resample=resample, **kwargs)
   5678 
-> 5679         im.set_data(X)
   5680         im.set_alpha(alpha)
   5681         if im.get_clip_path() is None:

~/anaconda3/lib/python3.7/site-packages/matplotlib/image.py in set_dat
a(self, A)
    683                 not np.can_cast(self._A.dtype, float, "same_ki
nd")):
    684             raise TypeError("Image data of dtype {} cannot be
 converted to "
--> 685                             "float".format(self._A.dtype))
    686 
    687         if not (self._A.ndim == 2

TypeError: Image data of dtype object cannot be converted to float
```



In [5]:

```
"""for scanning QR Code in Recorded Video"""
cap = cv2.VideoCapture('video2.mp4')
while (True):
# Read the frame
    check,img = cap.read()

    barcodes = pyzbar.decode(img)
    for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the
    # bounding box surrounding the barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on
    # our output image we need to convert it to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type
    # draw the barcode data and barcode type on the image
        text = "{} ({})".format(barcodeData, barcodeType)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)

    cv2.imshow('img', img)
    crop_img = img[y:y+h, x:x+w]

    # Stop if 'q' key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break
# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
```

```
[INFO] Found QRCODE barcode: 1@VwFvi2wCGh9q2Kd2t2m4WmXw3+LUtYEjO+sZr8y
TggPmXCkL/y1vfyMP,wZSU03WLkLDDj+zRPmL+HrZUIL2hLdUmNTz8if7wDiA=,xsKz6l1
JrM4J5tddfnw70g==
```

In [9]:

```
"""For Scanning QR Code during live videorecording"""
cap = cv2.VideoCapture(0)
while (True):
# Read the frame
    check,img = cap.read()
    barcodes = pyzbar.decode(img)
    for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the
    # bounding box surrounding the barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on
    # our output image we need to convert it to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type
    # draw the barcode data and barcode type on the image
        text = "{} ({})".format(barcodeData, barcodeType)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)
    cv2.imshow('img', img)
    # Stop if escape key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break
# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
```

In [1]:

```
#Files containing QR Codes
with open('datafile.txt') as file:
    qrlist=file.read().splitlines()
print(qrlist)
```

```
['111111', '111112', '111113', '111114', '111115', '111116', '111117',
'111118', '111119', '111120']
```

In [9]:

```
"""for checking Authorized or UnAutohetized entry"""
cap = cv2.VideoCapture(0)
while (True):
# Read the frame
    check,img = cap.read()
    barcodes = pyzbar.decode(img)
    for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the
    # bounding box surrounding the barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on
    # our output image we need to convert it to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

        if barcodeData in qrlist:
            myOutput = 'Authorized'
            myColor = (0,255,0)
        else:
            myOutput = 'Un-Authorized'
            myColor = (0, 0, 255)

    # draw the barcode data and barcode type on the image
        text = "{} ({}) {}".format(myOutput,barcodeType,barcodeData)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)
    cv2.imshow('img', img)
    # Stop if escape key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break
# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
```

In [ ]:

```
```