gnana ogy

(ted to J.N.T.U, Hyderabad)

Bach derabad, Telangan ia.

QR CODE READER AND GENERATOR

A course based project submitted in partial fulfilment of therequirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Dept. of Computer Science and Engineering

[redacted]

(Affiliated to J.N.T.U, Hyderabad)

Bachupally(v), Hyderabad, Telangana, India.

CERTIFICATE

This is to certify that [redacted] Nishanth [redacted]

have completed their course based project work at [redacted] Hyderabad entitled **"QR CODE READER and GENERATOR"** in partial fulfilment of the requirements for the award of B.Techdegree during the academic year 2020-2021. This work is carried out under my supervision andhas not been submitted to any other University/Institute for award of any degree/diploma.

CSE Department

DECLARATION

ACKNOWLEDGEMENT

Behind every achievement lies an unfathomable sea of gratitude to those who activated it, without which it would ever never have come into existence. To them I lay the words of gratitude imprinting within us.

Scheme of Course Based Project

Name of the course : Course Based project(Python laboratory)

Year / Semester : II-B.Tech II-semester

Project Title : QR·code Scanner and Reader

Done by :

Project Objectives : **The purpose of this project is to :**

- Explore and learn various modules in python like OpenCV, pyzbar , pyqrcode , etc and Tkinter library.
- Using the above libraries and modules to generate a QR code and Read it.

Project Outcomes : A output window having two options , one to generate and the other to scan are displayed. Once, after entering our message, a QR code is generated.

Description : In this project,we have provided a URL and a QR code is generated for that URL. This can be scanned using an imaging device like a camera.We have also created a scan qrcode button. When this button is clicked a camera screen is opened. When showed QR code on the camera then information of QR code is displayed on the Tkinter window

Stages involved:

1.Install the required modules ,libraries ,packages.
2.Use Tkinkter to design layout of the application.
3.Visualize the project.
4.Generating QR code when URL is entered.
5.Displaying the information of QR code when scan QR code is pressed.

ABSTRACT

QR code consists of black squares arranged in a square grid on a white background, which can be read by an imaging device such as a camera. By using Python, we can easily generate a QR code by installing a library called pyqrcode which has wide range of functions like make etc. This module attempts most of building process for creating QR codes.

We can read or scan the QR codes from the camera on our mobiles etc. To read the code, opencv library is used. After we run the code, in the output window we have to provide a URL and a QR code is generated for that URL. This can be scanned using an imaging device like a camera.

In This paper, a Quick Response (QR) Code Generator & Scanner, we are using Tkinter library for developing GUI interface. The advancement of technologies has brought about a huge change in mode of transactions. It has contributed to the increase in the range of services which a web applications can provide. The Quick Response (QR) Code technology can be regarded as a key technology in the future of mobile network operators in the telecommunication industry. This work is to present a model of mobile recharge solutions using the QR Code as the communication link, enabling encryption of transaction details using the python as well as their ability to be used in the exchange of financial value.

INDEX

Contents **PageNo**

LIST OF FIGURES

Contents **Page No.**

CHAPTER 1
INTRODUCTION

1.1 Introduction

The QR Code is the trademark for a type of matrix barcode (two-dimensional barcode) first designed by the automotive industry in Japan. Also, a barcode is a machine-readable optical label that contains information about the item to which it is attached. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to efficiently store data; extensions may also be used. It has a larger set of machine readable codes as compared to the one-dimensional barcode and it can hold more data because it uses both the horizontal and vertical axis. It is widely used in different fields such as manufacturing and mobile marketing. QR codes have a more advanced error correction mechanism and are more reliable as it has a faster speed than other codes (Adeel et al., 2014). Below is a sample of a QR Code.

1.1 Sample figure of a QR code

They were first created in 1994, its purpose was to track vehicles during manufacturing; it was also designed to allow high-speed component scanning. In 2002, when Japanese handset makers and others wanted to turn everyone's phone camera into a barcode scanner for marketing purposes, QR codes were very handy. With two dimensions of operation, QR codes are able to store several hundred times the amount of information carried by ordinary bar codes. They can contain anything that can fit into a maximum of about 4k (roughly one page of text). These codes are versatile. Application of QR Codes include their use on newspapers, magazines, journals, websites, advertisement, and advertisement board, where they are depleted to store websites' addresses, content information and miscellaneous data. Also, the QR Code is used in advertisements to guide people to visit their websites in the business world.

Additionally, the QR Code becomes an official tool that is used in the government and companies. In 2011, the Royal Dutch Mint announced that QR Code which will be embedded into the official coin would direct a user to a website about the Royal Mint’s centennial. In China, the QR Code is used on the train tickets on the corner of the right bottom and the names of passengers and relevant personal information are also included in this QR Code.

Before QR has been introduced payments are done through cash. After introducing QR code manual payment ,online transaction , internet transaction has been reduced.

During the pandemic situation, payments are done through QR code .

Before QR code introduced huge amount of notes were printed. After introduction of QR code, it has been reduced.

CHAPTER 2
LITERATURE

2.1 What is QR code?

QR codes are machine readable two-dimensional pixelated barcodes which can be used to store a variety of information. *QR* in *QR code* stands for *Quick Response*.

QR code was invented by a Japanese engineer *Masahiro Hara* from automobile manufacturer *Denso Wave* in the year 1994 to track the movement of car parts. QR Code has increased in popularity in the later 2010s with improvement in optical capabilities of mobile phones and their wide adoption. Nowadays, QR codes are being used for wide variety of applications like, make online payments, check hotel menu, share wifi password, obtain price and other details of products etc. QR Codes have become so popular that now every new smartphone comes with in built QR code reader**.**

2.2 Advanced QR code (Structure)

QR code can be customized using *QRCode* object which has the following parameters:

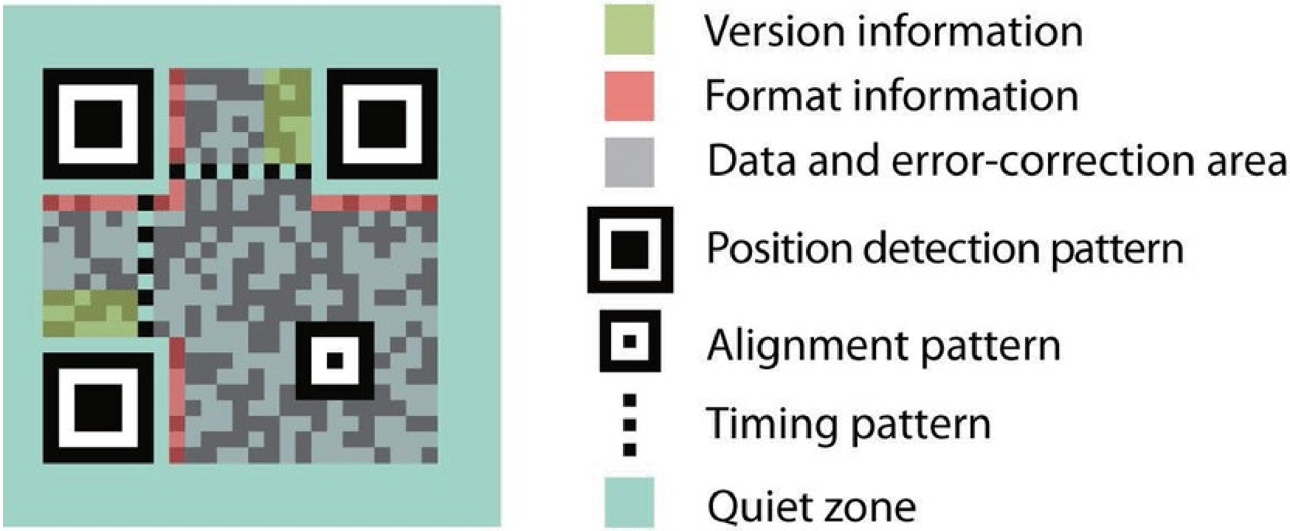


Fig2 structure of a QR code

2.2.1 Version:

There are 40 versions of QR code which controls the size of the code. 1 being the smallest and 40 being the largest. Version 1 will create a 21X21 matrix QR Code.

2.2.2 Error correction:

This parameter controls the Error Correction used for the QR code. This varies from 7%to30%,errorcorrection as below.
ERROR_CORRECT_L:upto:7%
ERROR_CORRECT_M:upto:15%
ERROR_CORRECT_Q:upto:25%

ERROR_CORRECT_H:upto30%

2.2.3 Box Size :
This parameter controls the number of pixels in each box of the QR code.

2.2.4 Border:

This parameter controls the thickness of the border. The default border is 4 pixels thick.

2.3 Modes of Transaction:

Let us take an example of **“Recharge”** for mobiles etc. Several pricing models have been postulated for services provided by MNOs. These include Pre-pay and Post-pay Pricing Model, Smart Data Pricing (SDP) model, Bundling Strategy based SDP, Micro-billing framework, Sealed-bid auction-based pricing, Sealed-bid reverse auction-based pricing, Cost-based pricing, Stackelberg game-based pricing, metered charging, Hybrid pricing model, Pay-asyou-go, Post-Paid and Pre-Paid. However, the most frequently used pricing model used in Nigeria is the prepaid and post-paid pricing models. For any of these pricing models, an exchange of cash for intended services is expected (Lu et al., 2017). Post-paid pricing model is an account-based model where subscribers are expected to deposit any amount into the MNOs bank account. However, with the pre-paid option, subscribers are expected to acquire the respective MNOs recharge vouchers. After acquisition, subscribers are expected to scratch the vouchers for the recharge codes. A lot of time is consumed in this process which is prone to error as the recharge code may have up to a sequence of sixteen numbers. Repeatedly entering wrong recharge pins three times may lead to the blockage of individual’s mobile line.

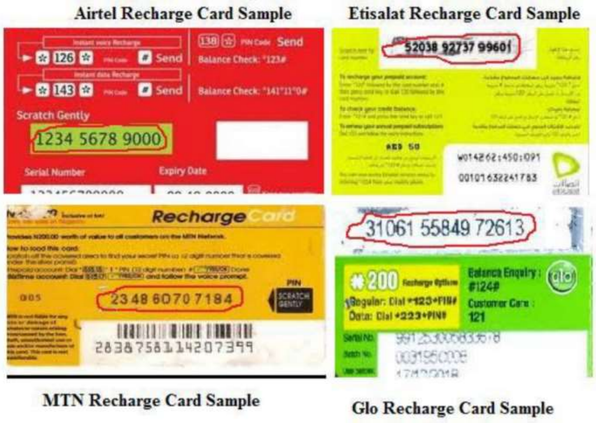


Fig2.3 Existing methods(example:recharge)

CHAPTER 3
REQUIREMENTS

We analyze, refine, and scrutinize the gathered requirements to make consistent and unambiguous requirements. Requirement analysis is significant and essential activity after elicitation.This activity reviews all requirements and may provide a graphical view of the entire system. After the completion of the analysis, it is expected that the understandability of the project may improve significantly.

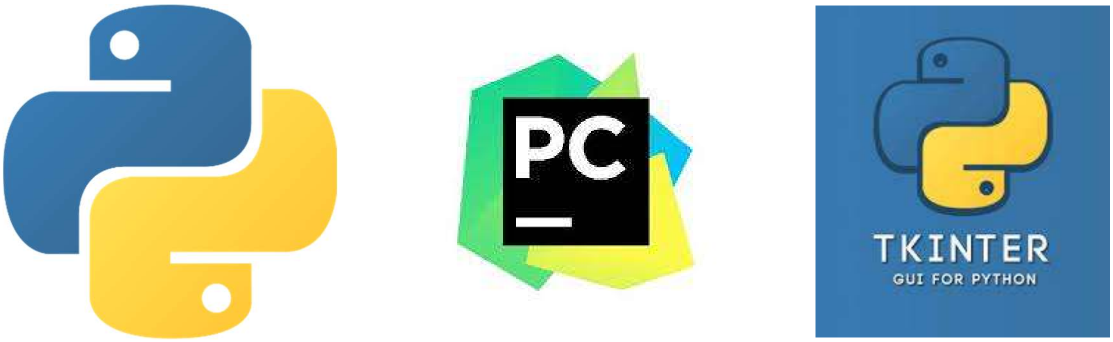


Fig 3.1 Requirements

3.1 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. The software requirements for this project are:

- **Python 3.10.2:**

  Python 3.10. 2 is the newest major release of the Python programming language, and it **contains many new features and optimizations**. This is a special bugfix release ahead of schedule to address a memory leak that was happening on certain function calls when using Cython

- **Pycharm:**

  PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

- **Opencv:**

  OpenCV-Python is a library of Python bindings designed to solve computer vision problems.OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

- **Pyzbar(module):**

It is a library in Pure python. It works with PIL/Pillow images, OpenCV/numpy.ndarrays , and raw bytes. It decodes locations of barcodes. NO dependencies other than zbar library itself.

- **Tkinter:**

  Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

- **Pyqrcode(module):**

  The pyqrcode module is a QR code generator that is simple to use and written in pure python. The module can automates most of the building process for creating QR codes. Most codes can be created using only two lines of code!

  Unlike other generators, all of the helpers can be controlled manually. You are free to set any or all of the properties of your QR code.

3.2 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list especially in case of operating systems.

- **Architecture** – All computer operating systems are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Although architecture-independent operating systems and applications exist, most need to be recompiled to run on a new architecture. See also a list of common operating systems and their supporting architectures.
- **Processing power** – The power of the central processing unit (CPU) is a fundamental system requirement for any software. Most software running on x86 architecture define processing power as the model and the clock speed of the CPU.
- **RAM**: 2GB and higher
- **Processor**: Intel i3 and above
- 20 MB of free disk space in the main memory of the device;
- Harddrive: Minimum 32 GB; Recommended 64 GB or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

CHAPTER 4
PROPOSED SOLUTION

In this project, we are generating QR codes which basically help in contactless payments etc. QR codes had been moving at a fairly steady pace, but they have really taken off in the last year. QR codes are everywhere now! They're easy to use and safe -- which make them a big draw for both consumers and businesses.

- QR codes promotes networking and sharing.

- QR codes can be linked to trailers, audio commentary, instructional videos or other media as a call-to-action. Or why not link your QR code to an 'Email us' or 'Call us' message where scanning the code lets people automatically email or call you.

- QR codes go one step further, letting you measure your results based on leads and clicks. You can use link-shortening services such as goo.gl to automatically generate QR codes that will let you share shortened links – they also give you useful analytics.

Fig4: Proposed solution

CHAPTER 5
MODEL IMPLEMENTATION

5.1 STAGES IN CODE GENERATION

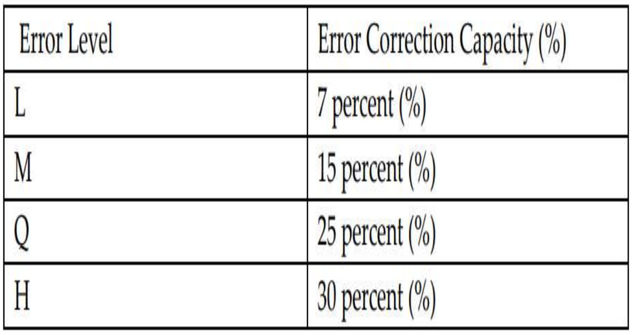
The QR Code generation goes through the following development processes:

5.1.1 DATA ANALYSIS

Usually, the QR code can be of four data types: numeric, alphanumeric, byte, and Kanji. These data types can be encoded into string of bits 1's and 0's in different ways. Hence, this stage analyzed the data to be encoded so as to determine its data type and the appropriate encoding mode to be adopted. The recharge PIN being considered here is of numeric data type hence; numeric encoding mode was adopted

5.1.2 DATA ENCODING

- QR Error Correction Levels: There are four levels of error correction depending on their error correction capabilities as shown in table. Error correction level L has the capability to recover 7 percent of damaged data while error correction level M could recover 15 percent of damaged data. Similarly, 25 percent and 30 percent of impaired data could be restored by error correction level Q and H respectively. With these conditions, one would have preferred error correction level H. However, the higher the correction level, the larger the bytes to be required as well as the size of the QR code.
- Determining the appropriate data version: QR code could be in different sizes called versions. Version 1 is of 21 × 21 pixel dimension while the next version is 4 levels higher than the previous version. Furthermore, each version has different character capacities which also depend on the encoding mode. The numeric mode and version 1 was adopted as it has 41-character capacity which is sufficient for the proposed task
- Determining the mode indicator: Each encoding mode has a four-bit mode indicator that identifies it. The encoded data must start with the appropriate mode indicator that specifies the mode being used for the bits that come after it. A sequence of 0's and 1's in four-bits are used to indicate the chosen encoding mode. For numeric encoding mode, 0001 is the mode indicator that will be used.
- Determining the character count indicator: Character count indicator specifies the number of characters to be encoded.
- Encoding the input data: the chosen encoding mode will be used to encode the input data. Different data encoding can be used by QR Code. Their complexity influences the amount of actual characters that can be stored inside the code.



Error Level	Error Correction Capacity (%)
L	7 percent (%)
M	15 percent (%)
Q	25 percent (%)
H	30 percent (%)

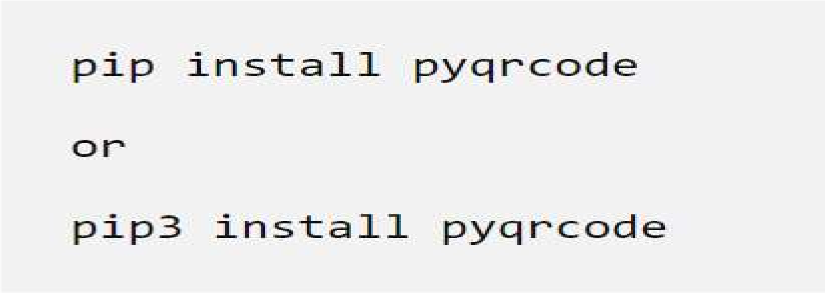
FIG 5.1 QR error correction levels

5.1.3 ERROR CORRECTION CODING:

Errors as a result of improper handling of the QR code may undermine the credibility of the data encoded in the QR code; therefore, when encoding the data, code words needed to correct the errors are also generated. Hence, both the encoded data and these code words will be read by the QR scanner and compared in order to ascertain if the correct data are decoded. As specified, a version 1 QR code with error correction level L can have up to 19 data code words in a single block, and only 7 error correction code words would need to be generated

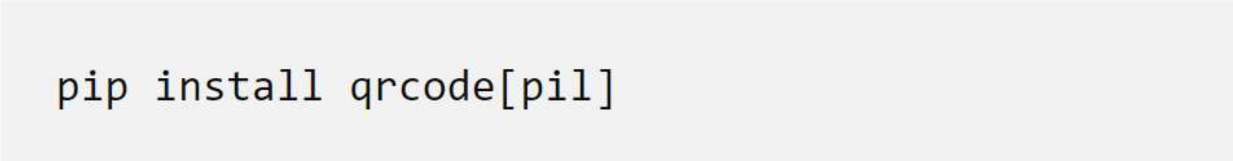
5.2 INSTALLATION OF PACAKGES

Python provides library **pyqrcode** which allows implementing QR code in Python Tkinter.
pyqrcode can be installed using pip or pip3 in the system. Use the following code



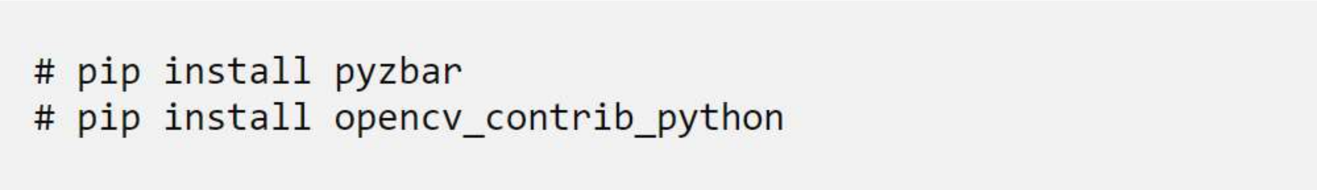
```
pip install pyqrcode

or

pip3 install pyqrcode
```

using **qrcode.make** method in **qrcode** library we can generate QR code.
And, using **qrcode.save** method in qrcode library we can save the generated QR code to the provided location.



```
pip install qrcode[pil]
```

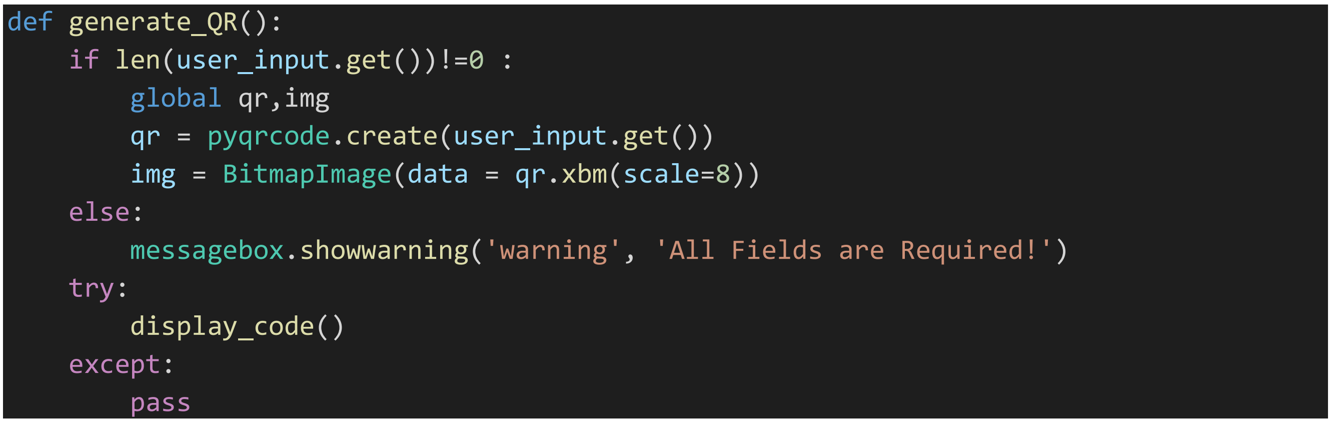
- Using the OpenCV library in python we will use our laptop’s camera to scan the QR code and the result will be printed on the Python Tkinter window.
- Other than OpenCV we also require pyzbar to handle the QR code recognition. Both the libraries can be installed using pip.



```
# pip install pyzbar
# pip install opencv_contrib_python
```

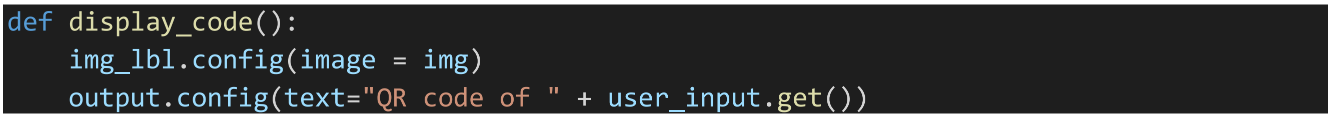
5.3 SOURCE CODE TO GENERATE QR CODE

Function to generate QR code:



```
def generate_QR():
    if len(user_input.get())!=0 :
        global qr,img
        qr = pyqrcode.create(user_input.get())
        img = BitmapImage(data = qr.xbm(scale=8))
    else:
        messagebox.showwarning('warning', 'All Fields are Required!')
    try:
        display_code()
    except:
        pass
```

In the above code pyrcode.create() function is used to create a QR code.
If the required fields are not filled by the user a function called as messagebox.showwarning() is used to warn the users.
Then userdefined function displaycode is written to display the QR code generated in the above code.

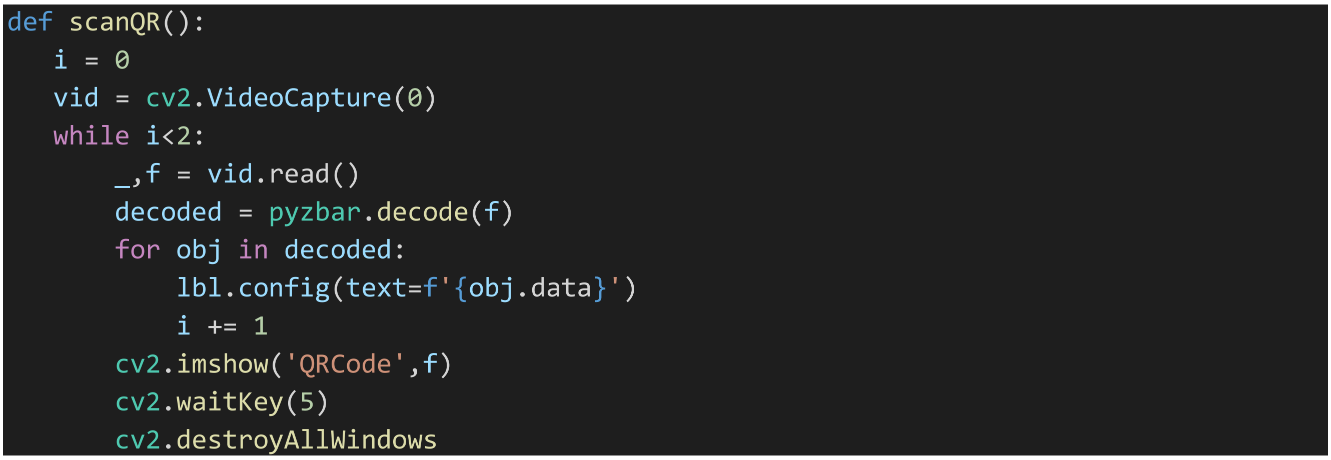


```
def display_code():
    img_lbl.config(image = img)
    output.config(text="QR code of " + user_input.get())
```

5.4 SOURCE CODE TO SCAN QR CODE

Using the OpenCV library in python we will use our laptop's camera to scan the QR code and the result will be printed on the Python Tkinter window.

- **vid = cv2.VideoCapture(0)** this code is to start the camera on the laptop.
- **while i<2:** the information of the QR code will be captured 2 times in a row
- **decoded = pyzbar.decode(f)**this code is to recognize and record the QRcode.
- the decoded variable will hold multiple information in a list format. Information like data, type, a rectangle property of image, width, height, etc.
- to fetch the data for loop is run and data will be displayed on the label.



```
def scanQR():
   i = 0
   vid = cv2.VideoCapture(0)
   while i<2:
       _,f = vid.read()
       decoded = pyzbar.decode(f)
       for obj in decoded:
           lbl.config(text=f'{obj.data}')
           i += 1
       cv2.imshow('QRCode',f)
       cv2.waitKey(5)
       cv2.destroyAllWindows
```

5.5 SOURCE CODE:

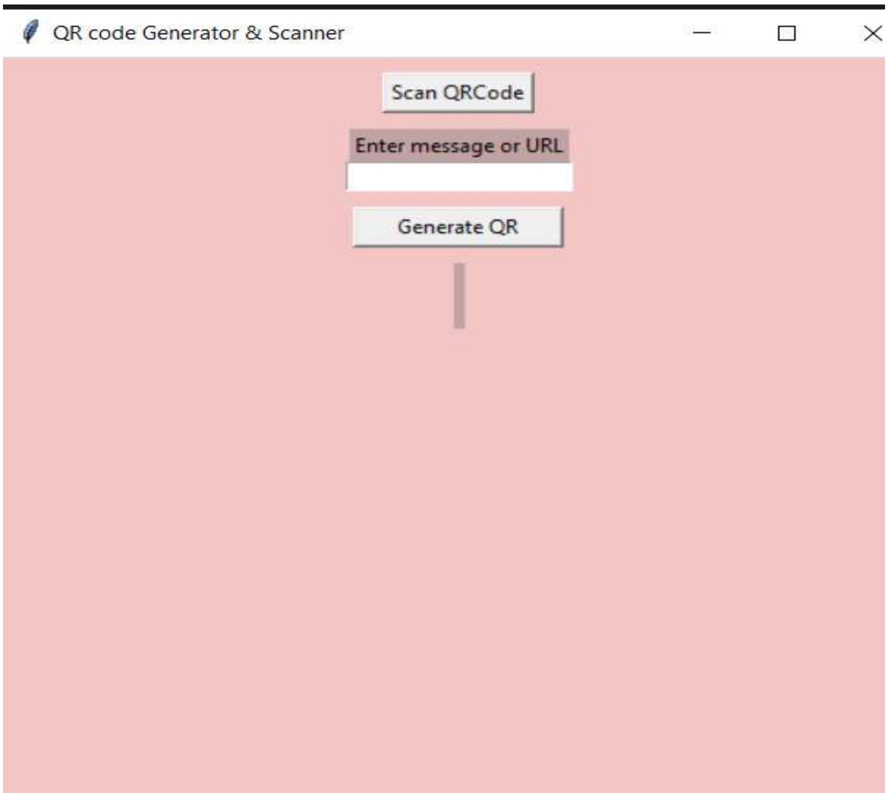
```
from tkinter import *
from tkinter import messagebox
import pyqrcode
import cv2
import pyzbar.pyzbar as pyzbar

def scanQR():
   i = 0
   vid = cv2.VideoCapture(0)
   while i<2:
      _,f = vid.read()
      decoded = pyzbar.decode(f)
```

```
        for obj in decoded:
            lbl.config(text=f'{obj.data}')
            i += 1
        cv2.imshow('QRCode',f)
        cv2.waitKey(5)
        cv2.destroyAllWindows

ws = Tk()
ws.title('QR code Generator & Scanner')
ws.geometry('500x500')
ws.config(bg='#F3C5C5')


Button(
    ws,
    text='Scan QRCode',
    command=scanQR
).pack(pady=10)




def generate_QR():
    if len(user_input.get())!=0 :
        global qr,img
        qr = pyqrcode.create(user_input.get())
        img = BitmapImage(data = qr.xbm(scale=8))
    else:
        messagebox.showwarning('warning', 'All Fields are Required!')
    try:
        display_code()
    except:
        pass

def display_code():
    img_lbl.config(image = img)
    output.config(text="QR code of " + user_input.get())


lbl = Label(
    ws,
    text="Enter message or URL",
    bg='#C1A3A3'
    )
lbl.pack()
```

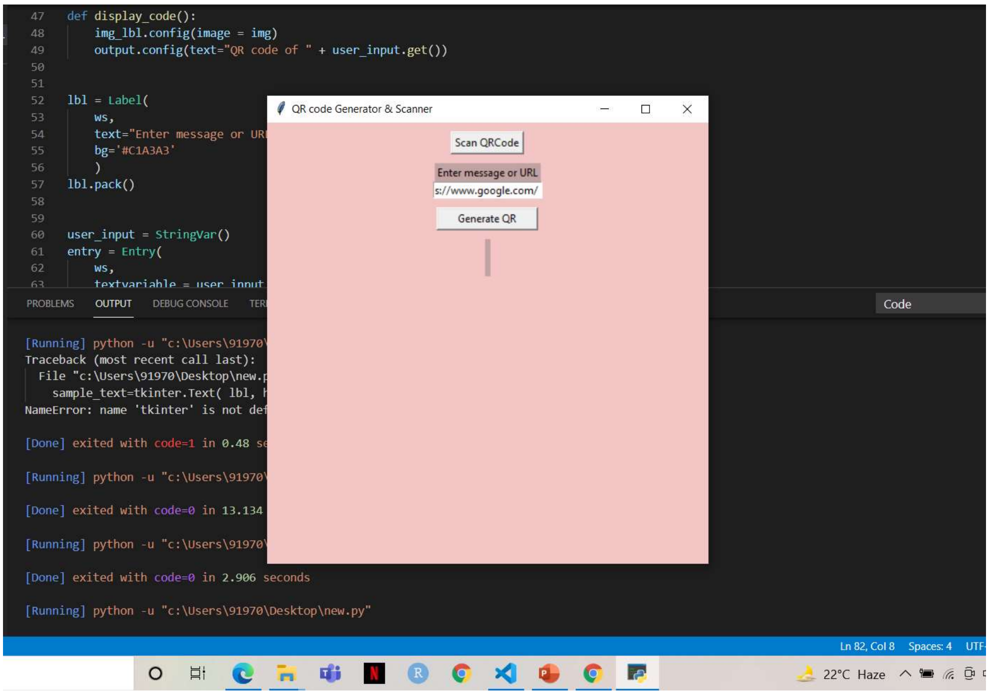
```
user_input = StringVar()
entry = Entry(
   ws,
   textvariable = user_input
   )
entry.pack(padx=10)


button = Button(
   ws,
   text = "Generate QR",
   width=15,
   command = generate_QR
   )
button.pack(pady=10)

img_lbl = Label(
   ws,
   bg='#C1A3A3')
img_lbl.pack()

output = Label(
   ws,
   text="",
   bg='#C1A3A3'
   )
output.pack()

ws.mainloop()
```

5.6 IMPLEMENTATION(OUTPUT)

In this output, we have provided a URL and a QR code is generated for that URL. This can be scanned using an imaging device like a camera.
The Python Tkinter window has a Scan QRCode button. When this button is clicked a camera screen is opened. When showed QR code on the camera then information of QR code is displayed on the Tkinter window.



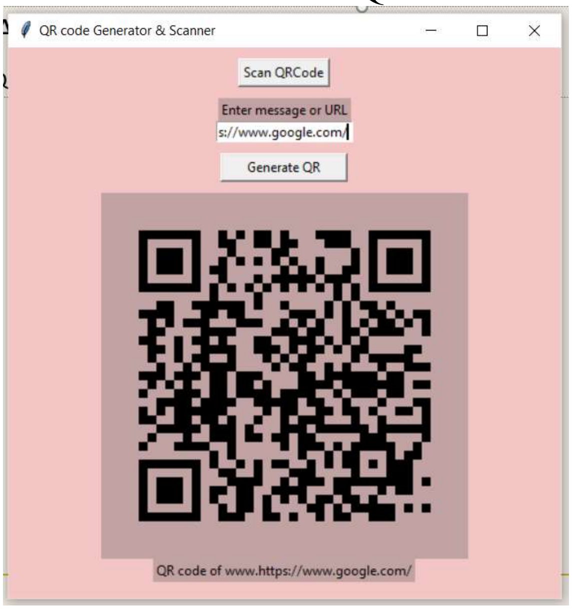
TYPING TEXT TO GENERATE QR CODE:



```
def display_code():
    img_lbl.config(image = img)
    output.config(text="QR code of " + user_input.get())


lbl = Label(
    ws,
    text="Enter message or URI
    bg='#C1A3A3'
    )
lbl.pack()


user_input = StringVar()
entry = Entry(
    ws,
    textvariable = user_input
```

```
[Running] python -u "c:\Users\91970\
Traceback (most recent call last):
  File "c:\Users\91970\Desktop\new.p
    sample_text=tkinter.Text( lbl, h
NameError: name 'tkinter' is not def

[Done] exited with code=1 in 0.48 se

[Running] python -u "c:\Users\91970\

[Done] exited with code=0 in 13.134

[Running] python -u "c:\Users\91970\

[Done] exited with code=0 in 2.906 seconds

[Running] python -u "c:\Users\91970\Desktop\new.py"
```

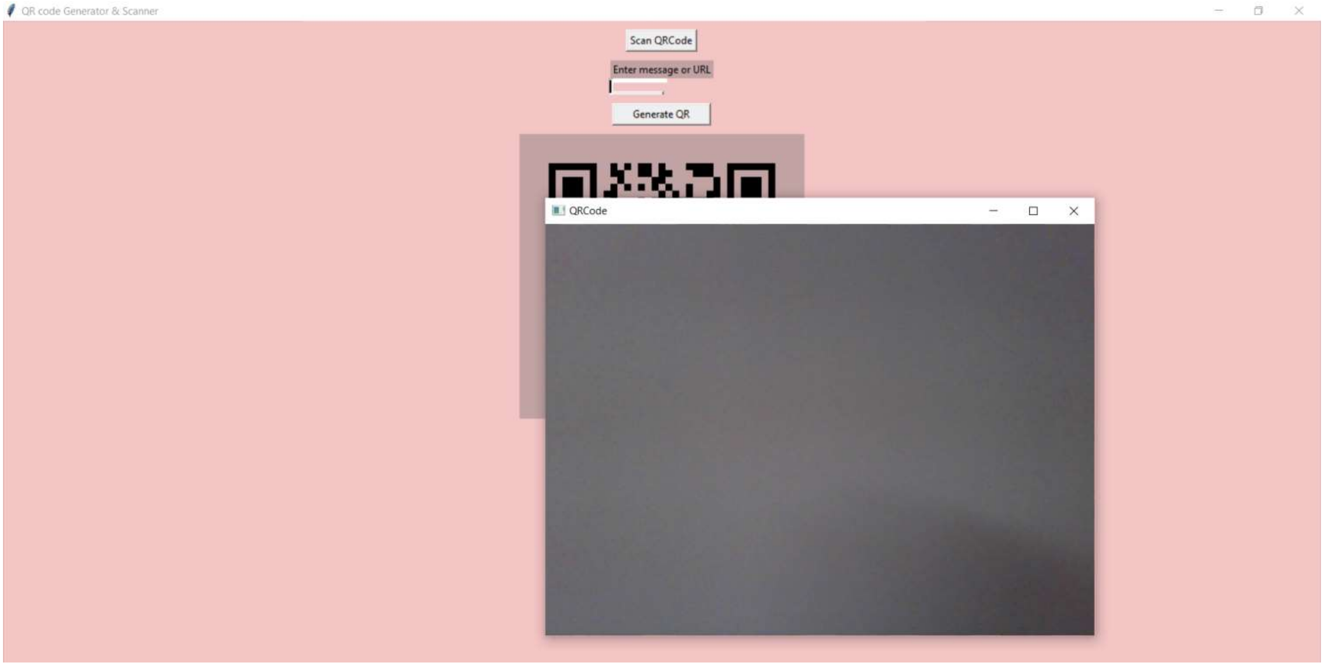
GENERATION OF QR CODE:



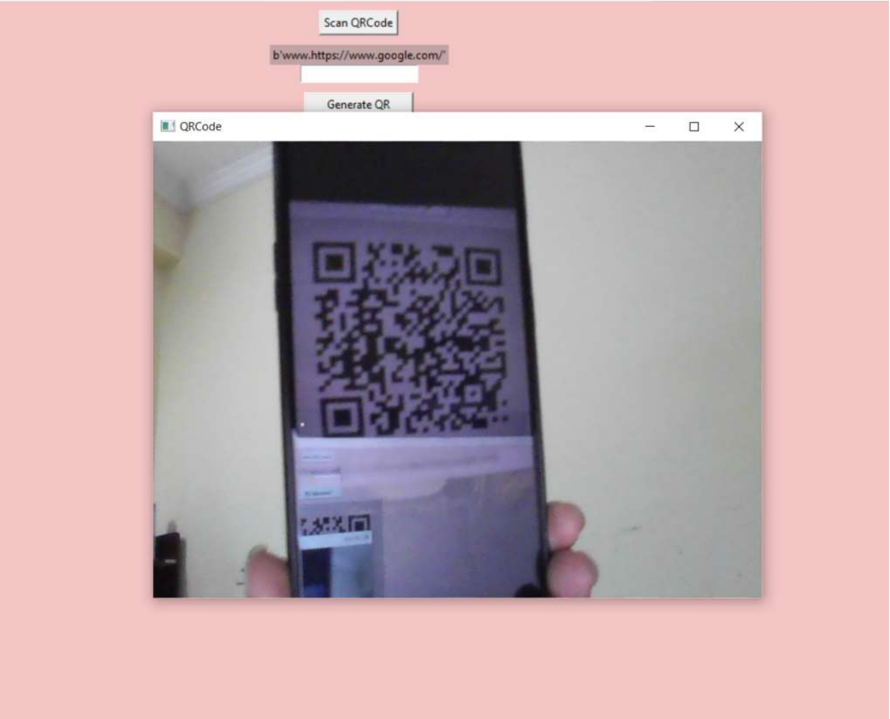
Enter message or URL
s://www.google.com/

QR code of www.https://www.google.com/

CLICKING ON SCAN QRCODE BUTTON:



READING THE QR CODE :



```
b'www.https://www.google.com/'
```

CHAPTER 6

TESTING

Testing errors:

QR code technology itself is basically bulletproof. Just look at how to scan a QR code for proof of its simplicity. But, very rarely, there may be an obstacle to scanning the code. That's trouble. And that's why we put together this QR code troubleshooting guide.

If you're unable to scan a QR code or have other general QR code scanning problems, we can help. And if you're here to see what QR code error correction is all about, we can help with that, too.

Read on, fix your problems, and scan away, scanners.

6.1 QR code Scanning Problems:

Learning how QR codes work will help you understand QR code issues. Any number of things can be happening when a QR code fails to yield its precious information. Through talking with QR code users and developers, we've isolated the five most common points of failure. Being aware of these should make it clear how to fix a QR code if it's being stubborn.

6.1.1Color and Contrast

A lot of businesses and brands value form over function. That leads them to some QR code marketing or QR code tracking experimentation for the sake of aesthetic unity with the rest of their branding. Often that means getting creative with their custom QR code's color or contrast.

QR codes were originally designed to have square black modules on a white background. That is, in turn, what most QR code scanners were designed to read. But any time a QR code flips that dynamic and places lighter squares over a darker background, it can trip some readers up.

Likewise, the QR code may have lighter colors over a darker background, but the difference between the two colors isn't big enough. There isn't enough contrast between background and foreground. Think of light blue modules over a dark blue background. This can also cause QR code scanners to fail.

The QR Code Is Too Complex

QR code uses are very numerous. From QR codes in healthcare to QR code food uses. Sometimes they're used to store quite a bit of information. And sometimes there's so much information encoded in it, it can't scan or scans extremely slowly.

This happens with static QR codes. If the phrase "static QR code" is stressing you out, don't let it. All it means is that the information you get by scanning the QR code is encoded directly in the black and white squares of the QR code itself. That's opposed to a dynamic QR code, which redirects to a website where the information is accessible.

All that means that the more complex a static QR code's information, the bigger and more complex the physical QR code has to be. More rows, more columns, more complicated patterns, etc. And if it gets too complicated it can take a long time for your QR code reader to scan it. And sometimes it can't be scanned at all.

6.1.2The QR Code Is Blurry or Crooked

Any optical device that scans QR codes requires precise sharpness and definition of the code. Any blurry squares or, crooked rows or columns, or otherwise shoddy formatting or arrangement will wreck the whole thing. That's why we recommend using a QR code template to properly situate QR codes on tables, windows, doors, and walls.

Your scanning device isn't great at filling in blanks and recreating parts of inaccessible code. Though it's getting better. See the QR Code error correction section below. But the most reliable way to scan a code is to make sure the code is in tip-top shape. The entire thing should be crystal clear and well-defined. If it's not, you'll have trouble scanning it.

6.1.3Size and Distance: QR Code Too Small?

Throughout QR code history, they've been smartly been standardized for international usage. That standardization is a size-to distance ratio.

For most current scanning devices, the distance-to-size relationship between a smartphone and its targeted QR code is 10:1. That means a QR code that’s 1 square inch requires the phone be 10 inches away.

So, is the QR code too small? Well, use the above ration. Extrapolate how large or small a QR code should be based on your distance from it. Or how far you should be from a QR code based on the QR code size. Either way, if you’re too far from the 10:1 ratio, you’ll have trouble scanning.

6.1.4The QR Code is Expired (or the Encoded URL Broken)

QR codes can expire, believe it or not. It’s a function of the business of QR code creation. Companies that professionally create QR codes put expiration dates on them or limit their number of scans. That, of course, generates more revenue for those companies. But it also leaves their customers with a bunch of arbitrary limits, inactive QR codes, and inaccessible QR code PDFs. Whether that’s because the QR code expired at a certain date, the business ran out of monthly scans, or PDF accessibility issues, the QR code will not be functional. And, yes, some companies actually require clients pay them *per scan* to use the QR codes they generate. Mercifully, that is not something we do with our digital restaurant menu. It’s a low flat free with no expiration and unlimited scans. That makes it a much cheaper, more agile menu strategy than building a menu app.

Or you generated a QR code for a URL and that QR code now redirects to a broken URL. We see this a lot when businesses use a free QR code generator online. It’s sad, but a lot of businesses depend on third-party, fly-by-night websites to create their QR codes for them. And that often involves that website using its own link for redirection. The QR code is effectively useless when that company goes under or can otherwise no longer maintain or guarantee the full operation of that link. Thankfully, we've got a resource to help you find the best QR code generator for you.

6.2 QR Code Error Correction: What Is It?

You know what a QR code is, but what is QRcode error correction? Conveniently, a QR code *does* have the ability to restore data to itself when it's damaged. It's called QR code error correction. It's a mathematical formula that is part of the QR code creation process. And what it does is convert the QR code into a polynomial that can be accessed if the code is damaged. In other words, it's a mathematical back-up of the visual QR code.

There are four levels of QR error code correction:

1. Level L: Able to backup 7% of the QR code's data
2. Level M: Able to backup 15% of the QR code's data
3. Level Q: Able to backup 25% of the QR code's data
4. Level H: Able to backup 30% of the QR code's data

The level of QR code error correction is chosen when the QR code is created. The reason why all QR codes aren't created with a level H error correction is because that makes the QR code bigger. The more data that's being backed up, the more space the QR code needs to devote to storing that data. That means a bigger code with longer load times. And it's doubly true for backing up complex QR codes. It's one of the best examples of the improvement of QR codes vs. barcodes.

If a QR code won't scan, it could be because the QR code error correction level is too low for the amount of damage the QR code has sustained. Or it could be that the QR code error correction level is so high—and the original code itself so complex—that the load times are inordinately long.

CHAPTER 7

CONCLUSION

7.1Conclusion

QR code is now being widely used in a variety of businesses. QR code is a way of encoding more information than a traditional bar code. And most importantly, it contains information that can be easily decoded at high speed. In this paper, we show how to create the QR codes via the web browser that facilitates users to easily create their own QR codes for websites, emails, business cards, print ads and so on. This platform could be used by different security conscience organizations. Text files or password system could be encrypted into QR Code and be read by a mobile device, etc. The work is achieved by the use of python flask framework which is the main interface for generating the QR Codes. The system will have a login access point to prevent unauthorized user to have access to the QR code generating session.

On the other hand, a mobile QR code scanner makes it easier to use mobile application to identify the encrypted code. This work has introduced a Quick Response (QR) Code generator with Mobile Scan Application for Mobile Network Recharge Operations. It has successfully demonstrated how QR codes can be used to secure Mobile Network recharge vouchers with a view to prevent unauthorized access to recharge pins that has characterized the existing recharge voucher patterns. The standalone software application developed was used to generate the QR code while the mobile application introduced was used to read the recharge code information embedded in form of QR code. The advent of smartphones with powerful features like mega pixel camera has also made the QR code scanning process easier contributing to its wide usage and acceptability.

7.2 Future Scope

The future of QR codes this year and beyond are not just simple guesses but are based on statistics. QR codes have become very handy in different aspects of people's lives and even in the business and marketing sector. With the current pandemic, the world is facing, QR codes have been very useful in delivering timely information in a form of contactless way, keeping everyone's safety as its main advantage.

CHAPTER 8
REFERENCES
BIBILOGRAPHY

[1] Lombardo, Nancy T, Anne Morrow, and Jeanne Le Ber. (2012). "Rethinking Mobile Delivery: Using Quick Response Codes to Access Information at the Point of Need." Medical Reference Services Quarterly. 31.1.

[2] Lou L., Tian Z., and Koh J. (2017). Tourist Satisfaction Enhancement Using Mobile QR Code Payment: An Empirical Investigation, 9(1186), 1–14. https://doi.org/10.3390/su9071186

[3] Probst, A. (2012). "The Expectations of Quick Response (QR) Codes in Print Media: An Empirical Data Research Anthology", UW-L Journal of Undergraduate Research XV, PP. (1-13)

[4] Sowern, S. (2011). "Beyond the simple codes: QR codes in education", Proceedings ascilite Hobart: Concise Paper, PP. (1157-1161).

[5] Sutheebanjard, P. and Premchaiswadi, W.(2010). "QR-Code Generator" In Proc. of IEEE, 8th International Conference on ICT and Knowledge Engineering.