```python
# Server Application (server.py)
# Bach Vu
# 01/08/2020

from request import *
from response import *
from socket import *
import sys, select

class DTServer():
    def __init__(self, hostname):
        self.sockets = [["English","Maori","German"], [None,None,None]]
        self.requests = [] # (byte_array, output_lang, sender_ip)
        self.hostName = hostname
        print("Server started with host name '{}'".format(hostname))

    def createSocket(self, ports):
        try:
            for i in range(3):
                sock = socket(AF_INET, SOCK_DGRAM)
                sock.bind((self.hostName, ports[i]))
                self.sockets[1][i] = sock
                print("Port {} is ready to receive {} requests".format(ports[i], self.sockets[0][i]))
            return True
        except Exception as e:
            raise e

    def shutdown(self):
        for socket in self.sockets[1]:
            socket.close()

    def getRequest(self):
        # get socket has buffer increase (new request)
        readable, _, _ = select.select(self.sockets[1], [], [])
        for sock in readable:
            option = -1
            if sock is self.sockets[1][0]:
                option = 0  # 0x0001 for English
            elif sock is self.sockets[1][1]:
                option = 1  # 0x0002 for Maori
            elif sock is self.sockets[1][2]:
                option = 2  # 0x0003 for German
            data, ip_sender = self.sockets[1][option].recvfrom(1024) # in byte
            self.requests.append( (data, option+1, ip_sender) )

    def sendResponse(self, response, target, s_ID):
        packet = response.encodePacket()
        if isinstance(packet, bytearray):
            socket = self.sockets[1][s_ID]
            socket.sendto(packet, target)
            print("Responded to sender at: {}".format(target))
        else:
            print("Respond failed with code {}! Try again ... ".format(packet))

####################### Main Program #################
def mainloop(server):
    while True:
        print("\nWaiting DT_request")
        server.getRequest()
        while len(server.requests) > 0:
            # Receive request
            packet = server.requests.pop(0)
            request = DT_Request.decodePacket(packet[0])
            if isinstance(request, int):
                err = "A request discarded with error Code {}:\n{}"
                err_mess = DT_Request.ErrorMessage[request-1]
                print(err.format(int(request), err_mess))
                continue
            print(request)

            # Reply
            print("Preparing response in {}.".format(server.sockets[0][packet[1]-1]))
            response = DT_Response(packet[1], request.requestType)
            server.sendResponse(response, packet[2], packet[1]-1)

def checkInputArgv():
    if len(sys.argv) != 4:
        return 1

    ports = []
    try:
        ports = [int(sys.argv[1]), int(sys.argv[2]), int(sys.argv[3])]
    except BaseException:
        return 2

    for port in ports:
        if port < 1024 or port > 64000:
            return 3
    return 0

def startServer():
    print("\nWelcome to DT Finder (Server)")
    # Error Checking
    errMess = [
        "Argument input error.\n    python server.py {host} {port_eng} {port_maori} {port_ger}",
        "Ports input must be integer (whole number).",
        "Port must be between 1024 and 64000 inclusively!"
    ]
    errCode = checkInputArgv()
    if errCode != 0:
        print(errMess[errCode-1])
        sys.exit()

    # Create Server instance
    host = getfqdn()
    ports = [int(sys.argv[1]), int(sys.argv[2]), int(sys.argv[3])]
    server = DTServer(host)
    server.createSocket(ports)
    return server

if __name__ == "__main__":
    try:
        DT_server = startServer()
        mainloop(DT_server)
    except KeyboardInterrupt:
        DT_server.shutdown()
        print("Program exited!")
    except Exception as e:
        print(e)
        print()
```