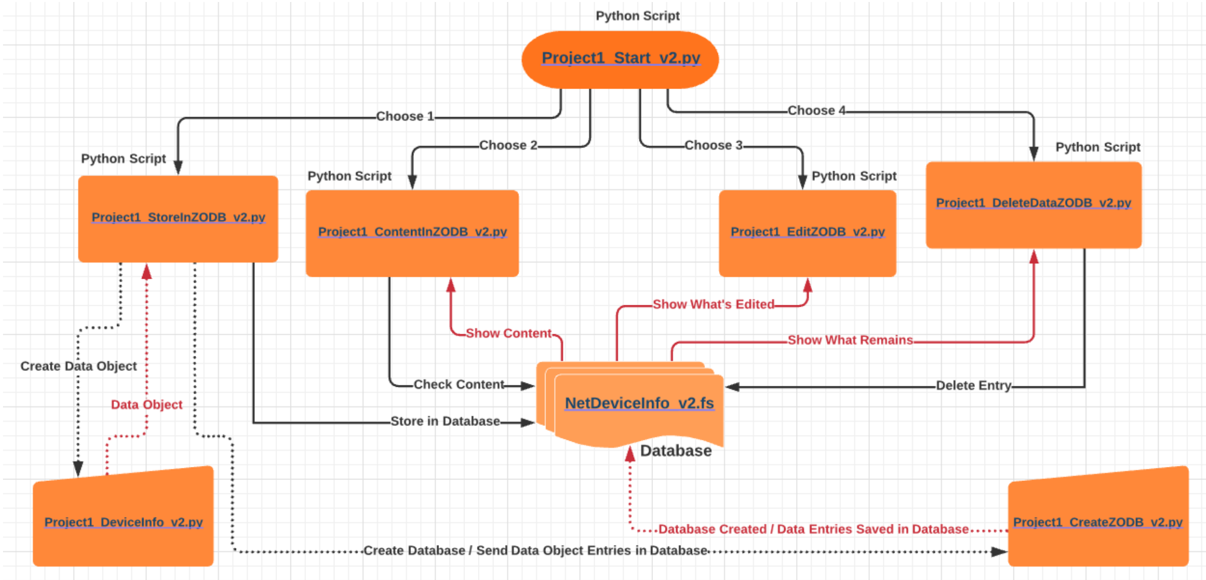
Data Flow Chart of this Project is shown bellow:



Content of the Python scripts used for this Project example is shown bellow:

Project1_DeviceInfo_v2.py

```
#!/usr/bin/env python3


#persistance package forms the core protocol for making objects interact “transparently” with a database such as the ZODB
from persistent import Persistent


#Using this class as blueprint we are able later to create objects, that have arguments:ID, IP_address and netmask
class DeviceInfo(Persistent):
    def __init__(self,ID,IP_address,netmask):
        self.ID=ID
        self.IP_address=IP_address
        self.netmask=netmask
        print("=== Device data card ===")
        print("ID: ",self.ID)
        print("Device IP address: ", self.IP_address)
        print("Netmask: ", self.netmask)
```

Project1_CreateZODB_v2.py

```
#!/usr/bin/env python3

# We are creating our database

from ZODB import FileStorage, DB
import transaction

#We are setting up connection to the database
class CreateZODB(object):
   def __init__(self, ZODB_file_path):
      self.storage = FileStorage.FileStorage(ZODB_file_path) #we create storage object
      self.db = DB(self.storage)    #storage object is linked with database object
      self.connection = self.db.open()   #we are establishing connection to the database
      self.dbroot = self.connection.root()   #we can access 'root' of the database using '.root()' method

   def close(self):     #closing of the connection to the database
      self.connection.close()
      self.db.close()
      self.storage.close()
```

Project1_StoreInZODB_v2.py

```
#!/usr/bin/env python3


# We need to store some data into our ZODB

from Project1_DeviceInfo_v2 import DeviceInfo

dev=[]
n = int(input("Enter number of devices: "))

#We create empty lists
dev_ID = []
dev_IP_address = []
dev_netmask = []

for i in range(1, n+1):
   print("Enter device",i,"ID: ")
   elem_ID = str(input())
   print("Enter device",i,"IPaddress: ")
   elem_IP_address = str(input())
   print("Enter device",i,"netmask: ")
   elem_netmask = str(input())

   dev_ID.append(elem_ID)
```

```
    dev_IP_address.append(elem_IP_address)
    dev_netmask.append(elem_netmask)

#We are printing to see what form of data we have
print(dev_ID)
print(dev_IP_address)
print(dev_netmask)


from Project1_CreateZODB_v2 import CreateZODB, transaction
db = CreateZODB('./NetDeviceInfo_v2.fs')
dbroot = db.dbroot

for i in range(0, len(dev_ID)):
    print(dev_ID[i])
    # use globals() to store the vars in the global stack
    globals()["dev" + str(i)] = DeviceInfo(dev_ID[i],dev_IP_address[i],dev_netmask[i]) #we put entered values from
                                                                                        # keyboard as arguments for
                                                                                        # "DeviceInfo()" class

    dbroot['Details of Device'+ str(i)] = globals()["dev" + str(i)]     #we mark and store these data inside database

transaction.commit()
db.close()
```

Project1_ContentInZODB_v2.py

```
#!/usr/bin/env python3

# Show saved content in the database

from Project1_CreateZODB_v2 import CreateZODB
db = CreateZODB('./NetDeviceInfo_v2.fs')
dbroot = db.dbroot

from Project1_DeviceInfo_v2 import DeviceInfo
for key in dbroot.keys():
    obj = dbroot[key]
    if isinstance(obj, DeviceInfo):   #we check if object is linked with class - "isinstance()" returns True if is.
        print("Device:", obj.ID)
        print("Device IP address:", obj.IP_address, "Netmask:", obj.netmask)

db.close()
```

Project1_EditZODB_v2.py

```
#!/usr/bin/env python3

from Project1_CreateZODB_v2 import CreateZODB, transaction
db = CreateZODB('./NetDeviceInfo_v2.fs')
dbroot = db.dbroot


#User enters which device data will be edited
DevNum = str(input("Enter device number: "))

#User enters new device data
DevID = str(input("Enter new device ID:"))
DevIP_address = str(input("Enter new device IP address:"))
DevNetmask = str(input("Enter new device netmask:"))

from Project1_DeviceInfo_v2 import DeviceInfo

#We create object that points to marked device data in our database
DevObject = dbroot['Details of Device' + str(DevNum)]

#We are storing changes in database
DevObject.ID = DevID
DevObject.IP_address = DevIP_address
DevObject.Netmask = DevNetmask

#We confirm transaction of newest changes and we close the database
transaction.commit()
db.close()
```

Project1_DeleteDataZODB_v2.py

```
#!/usr/bin/env python3


from Project1_CreateZODB_v2 import CreateZODB, transaction
db = CreateZODB('./NetDeviceInfo_v2.fs')
dbroot = db.dbroot


#User enters which device data will be deleted
DevNum = str(input("Enter device number: "))

from Project1_DeviceInfo_v2 import DeviceInfo

#We delete marked object in database
del dbroot['Details of Device' + str(DevNum)]
```

```
#We confirm transaction of newest changes and we close the database
transaction.commit()
db.close()
```

Project1_Start_v2.py

```
#!/usr/bin/env python3

i=1
while i<1000:
    print("                  INSTRUCTIONS                    ")
    print("---------------------------------------------------------")
    print("Enter number from the keyboard: \n")
    print("0 - if you need to exit this screen")
    print("1 - if you need to insert data into the database")
    print("2 - if you need to check data content inside the database")
    print("3 - if you need to edit data from the database")
    print("4 - if you need to remove data from the database")

    number = int(input("Enter a number: "))

    if number<0:
        print("You need to enter number from 1 to 4!")
        break

    elif number>4:
        print("You need to enter number from 1 to 4!")
        break

    elif number==0:
        print("Thank you.")
        break

    else:
        #We execute Python Script 1 to store into the database
        if number == 1:
            exec(open('.//Project1_StoreInZODB_v2.py').read())

        #We execute Python Script 2 to check what data are inside database
        elif number == 2:
            exec(open('.//Project1_ContentInZODB_v2.py').read())

        #We execute Python Script 3 to edit stored data inside database
        elif number == 3:
            exec(open('.//Project1_EditZODB_v2.py').read())

        #We execute Python Script 4 to delete data stored inside database
        elif number == 4:
            exec(open('.//Project1_DeleteDataZODB_v2.py').read())
    i+=1
```