```python
from data_class import Data
from loadingModule_2 import loadingAnimation
import nltk

# method that returns the question-type like 'what', 'when', where' etc...
def printAllQuestionType(processed_data):
    if len(processed_data.get("QT")) == 1:
        return "'" + processed_data.get("QT").__getitem__(0) + "'"
    else:
        result = "mix of "
        for c in range(len(processed_data.get("QT")) - 1):
            result += "'" + processed_data.get("QT").__getitem__(c) + "', "
        result += "and '" + processed_data.get("QT").__getitem__(len(processed_data.get("QT")) - 1) + "'"
    return result

# method that returns any possible places that the user mentioned in their original statement ~ improves specific contextualization
def printPossiblePlaceNames(saved_input):
    result = ""
    for places in data_instance.specificPlaceList:
        if places.lower() in saved_input.lower():
            result += places + ", "
    return result

# introductory instructions and terms/conditions
print("QUESTION ANALYZER: \n1.) Enter a question that you might ask a Google Assistance, Alexa, Siri, Cortana, etc..."
      "\n2.) The system will process your response.\n3.) It will give you it's understanding of the question by categorizing it using basic Natural Language Processing (NLP) algorithm.\n")
print("This is a first step taken to complete a part of 'robotics data interpretation' and will be built upon modularly.\n")
print("This model is still in BETA; some questions might not be recognizable by the system. More updates will be rolling out soon. [Version: 1.3]\n\n")
user_input = input("Ask me anything: ").lower()
data_instance = Data()
rowCategory = None

# loop helps the user continue asking more questions for system categorization
while user_input != "stop":
    saved_input = user_input
    processed_data = data_instance.parsedData(data_instance.stemWord(user_input))

    # ensures there are no error, else, redirect the issue
    if processed_data.get("I") != 'ERROR':
        loadingAnimation()
        for row in range(len(data_instance.possibleList)):
            for column in range(len(data_instance.possibleList[row])):
                for a in range(len(processed_data.get("I"))):
                    if data_instance.possibleList[row][column] == processed_data.get("I").__getitem__(a):
                        rowCategory = str(row)
        print(f"\n\n\nI understand that you are trying to ask a question that starts with a {printAllQuestionType(processed_data)} and I am supposed to give a(n) ")

        # list of possible type of questions extracted from user input [non-exhaustive]
        if rowCategory == "0":
            print("concise and short answer for general information related to places, conditions, activities, and forecasts.")
        elif rowCategory == "1":
            print("personal/health based information regarding diet/lifestyle.")
        elif rowCategory == "2":
            print("productivity tailored response for improving user efficiency.")
        elif rowCategory == "3":
            print("entertainment related answer that is tailored to improving user's leisure.")
        elif rowCategory == "4":
            print("mathematical answer, which can either be short or informative, for a concept or calculation.")
        elif rowCategory == "5":
            print("response that improves the user's knowledge in crucial details regarding current events.")
        elif rowCategory == "6":
            print("advice to the user in the best manner possible that will give them better guidance.")
        if (printPossiblePlaceNames(saved_input) != ""):
            print("You also mentioned " + printPossiblePlaceNames(saved_input) + "meaning you want
my responses to be personalized for that/those place(s).")
        user_input = input("\nIs my understanding right? Type 'Y' for Yes or 'N' for No: ")
        print("Glad I am doing it right. Data has been noted!") if user_input == "Y" else input("In what way should I have interpreted the response: ")
        print("Thank you for your feedback!\n")
        user_input = input("\nAsk me anything (or type 'stop' to end): ")
    else:
        user_input = input("\nThis question is unrecognizable. Please note that this model is in its early stages. Try again or type 'stop': ")
```