The project is programmed with python programming ,langage
python version:3.8
:libraries used
- json
- operator
- glob
- matplotlib(must be downloaded)

the program gets input data from json files with spacific -1 :structure


```
                                    }
                        ,"name": "john duo"
                ,"mobileNumber": "051612651500O"
                        ,"email": "john@gmail.com"

                                ] :"books"
                                    }
                    ,"bookTitle": "vampire diaries"
                                , pages": 764"
                        "category":"super natural"
                                                ,{
                                                }
                    ,"bookTitle": "the fall"
                                , pages": 987"
                            "category":"horror"
                                                ,{
                                                }
                ,"bookTitle": "the night eternal"
                                , pages": 155"
                        "category":"super natural"
                                                ,{
```

```
                                                    }
                                ,"bookTitle": "the convoluted universe"
                                                , pages": 966"
                                            "category":"astrology"
                                                                ,{
                                                                }
                                ,"bookTitle": "rangers"
                                                , pages": 317"
                                            "category":"documentary"
                                                                {
                                                                   [
                                                                     {
```

the json files must be at the same folder as the program -2
script (readingGroups.py)
the program is set to 3 groups for 20 reader for each, -3
two test the 3 groups are working lower the limit of each
group to 4 to match the 12 exicting json files
or increase the number of json files in folder

File1 (.json) user1


```
{
    "name": "Ahmed Osman",
    "mobileNumber": "54986518",
    "email": "Ahmed@gmail.com",

    "books": [
        {
            "bookTitle": "murder on the orient express",
            "pages": 255 ,
            "category":"crime"
        },
        {
            "bookTitle": "harry potter",
            "pages": 850 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "the kill order",
            "pages": 325 ,
            "category":"crime"
        },
        {
            "bookTitle": "escape room",
            "pages": 200 ,
            "category":"horror"
        },
        {
            "bookTitle": "train to bosan",
            "pages": 430 ,
            "category":"horror"
        },
        {
            "bookTitle": "running with scissors",
            "pages": 365,
            "category": "biography"
        }
    ]
}
```

File2 (.json) user2


```
{
    "name": "Ali Ebrahim",
    "mobileNumber": "0000",
    "email": "Ali@gmail.com",

    "books": [
        {
            "bookTitle": "come away",
            "pages": 50 ,
            "category":"adventure"
        },
        {
            "bookTitle": "train to bosan",
            "pages": 430 ,
            "category":"horror"
        },
        {
            "bookTitle": "The Prisoner of zinda",
            "pages": 100 ,
            "category":"thrill"
        },
        {
            "bookTitle": "escape room",
            "pages": 200 ,
            "category":"horror"
        },
        {
            "bookTitle": "WW1",
            "pages": 300 ,
            "category":"history"
        },
        {
            "bookTitle": "the kill order",
            "pages": 325 ,
            "category":"crime"
        },
        {
            "bookTitle": "2012",
            "pages": 1050 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "undaunted courage",
            "pages": 399 ,
            "category":"action"
        }
    ]
}
```

File3 (.json) user3


```
{
    "name": "Ebrahim Nagy",
    "mobileNumber": "05161265000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "book of eli",
            "pages": 175 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "2012",
            "pages": 1050 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "extraction",
            "pages": 155 ,
            "category":"action"
        },
        {
            "bookTitle": "invisible man",
            "pages": 750 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "unchanged",
            "pages": 380 ,
            "category":"comedy"
        },
        {
            "bookTitle": "the kill order",
            "pages": 325 ,
            "category":"crime"
        }
    ]
}
```

File4 (.json) user4


```
{
    "name": "Mohamed Zaki",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "Broken Glass",
            "pages": 350 ,
            "category":"literature"
        },
        {
            "bookTitle": "the girl with the dragon tatto",
            "pages": 425 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "no treason",
            "pages": 155 ,
            "category":"non fiction"
        },
        {
            "bookTitle": "vangard",
            "pages": 72 ,
            "category":"action"
        },
        {
            "bookTitle": "The Prisoner of zinda",
            "pages": 100 ,
            "category":"thrill"
        }
    ]
}
```

File5 (.json) user5


```
{
    "name": "Mansour Ahmed",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "Green land",
            "pages": 125 ,
            "category":"action"
        },
        {
            "bookTitle": "echo boomers",
            "pages": 665 ,
            "category":"drama"
        },
        {
            "bookTitle": "rangers",
            "pages": 317 ,
            "category":"documentary"
        },
        {
            "bookTitle": "clachinkof",
            "pages": 454 ,
            "category":"war"
        },
        {
            "bookTitle": "war zone",
            "pages": 784 ,
            "category":"war"
        }
    ]
}
```

File6 (.json) user6


```
{
    "name": "Hazem Osama",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "come away",
            "pages": 50 ,
            "category":"adventure"
        },
        {
            "bookTitle": "train to bosan",
            "pages": 430 ,
            "category":"horror"
        },
        {
            "bookTitle": "check fight",
            "pages": 170 ,
            "category":"action"
        },
        {
            "bookTitle": "the broken hearts",
            "pages": 580 ,
            "category":"drama"
        },
        {
            "bookTitle": "the broken hearts",
            "pages": 580 ,
            "category":"drama"
        }

    ]
}
```

File7 (.json) user7


```
{
    "name": "Said Mohamed",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "Broken Glass",
            "pages": 190,
            "category":"literature"
        },
        {
            "bookTitle": "the girl with the dragon tatto",
            "pages": 215 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "no treason",
            "pages": 275 ,
            "category":"non fiction"
        },
        {
            "bookTitle": "vangard",
            "pages": 430 ,
            "category":"action"
        },
        {
            "bookTitle": "the girl with the dragon tatto",
            "pages": 425 ,
            "category":"science fiction"
        }
    ]
}
```

File8 (.json) user8


```
{
    "name": "Mohamed Hassan",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "color me",
            "pages": 200 ,
            "category":"fashion"
        },
        {
            "bookTitle": "interview guide",
            "pages": 414 ,
            "category":"careers"
        },
        {
            "bookTitle": "death note",
            "pages": 515 ,
            "category":"comic books"
        },
        {
            "bookTitle": "discrete math",
            "pages": 318 ,
            "category":"programming"
        },
        {
            "bookTitle": "Broken Glass",
            "pages": 350 ,
            "category":"literature"
        }
    ]
}
```

File9 (.json) user9


```
{
    "name": "Hesham Nabil",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "naruto",
            "pages": 529 ,
            "category":"comic books"
        },
        {
            "bookTitle": "vampire knight",
            "pages": 427 ,
            "category":"comic books"
        },
        {
            "bookTitle": "think theory",
            "pages": 677 ,
            "category":"brain games"
        },
        {
            "bookTitle": "running with scissors",
            "pages": 365,
            "category": "biography"
        },
        {
            "bookTitle": "rangers",
            "pages": 317 ,
            "category":"documentary"
        }
    ]
}
```

File10 (.json) user10


```
{
    "name": "abdelrahman hossam",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "vampire diaries",
            "pages": 764 ,
            "category":"super natural"
        },
        {
            "bookTitle": "the fall",
            "pages": 987 ,
            "category":"horror"
        },
        {
            "bookTitle": "the night eternal",
            "pages": 155 ,
            "category":"super natural"
        },
        {
            "bookTitle": "the convoluted universe",
            "pages": 966 ,
            "category":"astrology"
        },
        {
            "bookTitle": "rangers",
            "pages": 317 ,
            "category":"documentary"
        }
    ]
}
```

File11 (.json) user11


```
{
    "name": "Patrick Stone",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "john adams",
            "pages": 405 ,
            "category":"military"
        },
        {
            "bookTitle": "undaunted courage",
            "pages": 399 ,
            "category":"action"
        },
        {
            "bookTitle": "night",
            "pages": 155 ,
            "category":"romance"
        },
        {
            "bookTitle": "1776",
            "pages": 500 ,
            "category":"history"
        },
        {
            "bookTitle": "invisible man",
            "pages": 750 ,
            "category":"science fiction"
        }
    ]
}
```

File12 (.json) user12


```
{
    "name": "Mohamed Hassan",
    "mobileNumber": "0516126515000",
    "email": "Ebrahim@gmail.com",

    "books": [
        {
            "bookTitle": "Broken Glass",
            "pages": 500 ,
            "category":"literature"
        },
        {
            "bookTitle": "the girl with the dragon tatto",
            "pages": 600 ,
            "category":"science fiction"
        },
        {
            "bookTitle": "no treason",
            "pages": 155 ,
            "category":"non fiction"
        },
        {
            "bookTitle": "vangard",
            "pages": 500 ,
            "category":"action"
        },
        {
            "bookTitle": "train to bosan",
            "pages": 430 ,
            "category":"horror"
        }
    ]
}
```

The programing code

```
import json
import operator
from glob import glob
import matplotlib.pyplot as plt


class Reader:
    """
    this class declares readers that we will do our calculations
on
    """
    def __init__(self, name, mobile_number, email, no_books,
no_pages, category):
        """
        This is a constructor that assigns the values of the
parameters to the attributes of the instance
        :param name:
        :param mobile_number:
        :param email:
        :param no_books:
        :param no_pages:
        :param category:
        """
        self.name = name
        self.mobile_number = mobile_number
        self.email = email
        self.no_books = no_books
        self.no_pages = no_pages
        self.category = category

    def get_name(self):
        """
        An accessor
        :return: name of reader
        """
        return self.name

    def get_mobile_number(self):
        """
        An accessor
        :return: mobile number of the reader
        """
        return self.mobile_number

    def get_email(self):
        """
        An accessor
        :return: reader's E-mail
```

```
        """
        return self.email

    def get_no_books(self):
        """
        An accessor
        :return: number of books
        """
        return self.no_books

    def get_no_pages(self):
        """
        An accessor
        :return: number of pages per book
        """
        return self.no_pages

    def get_category(self):
        """
        An accessor
        :return: the category of the book
        """
        return self.category


class ReadingGroup:
    """
    This class groups readers in  a single instance
    """
    def __init__(self, name, max_num):
        self.name = name
        self.max_num = max_num
        self.readers = []

    def add_reader(self, reader):
        if len(self.readers) < self.max_num:
            self.readers.append(reader)
            return True
        return False

    def get_total_books(self):
        total_books = 0
        for reader in self.readers:
            total_books = total_books + reader.get_no_books()

        return "Total Number of books for {} is {}
book".format(self.name, total_books)

    def get_total_pages(self):
        total_pages = 0
        for reader in self.readers:
            total_pages = total_pages + reader.get_no_pages()
```

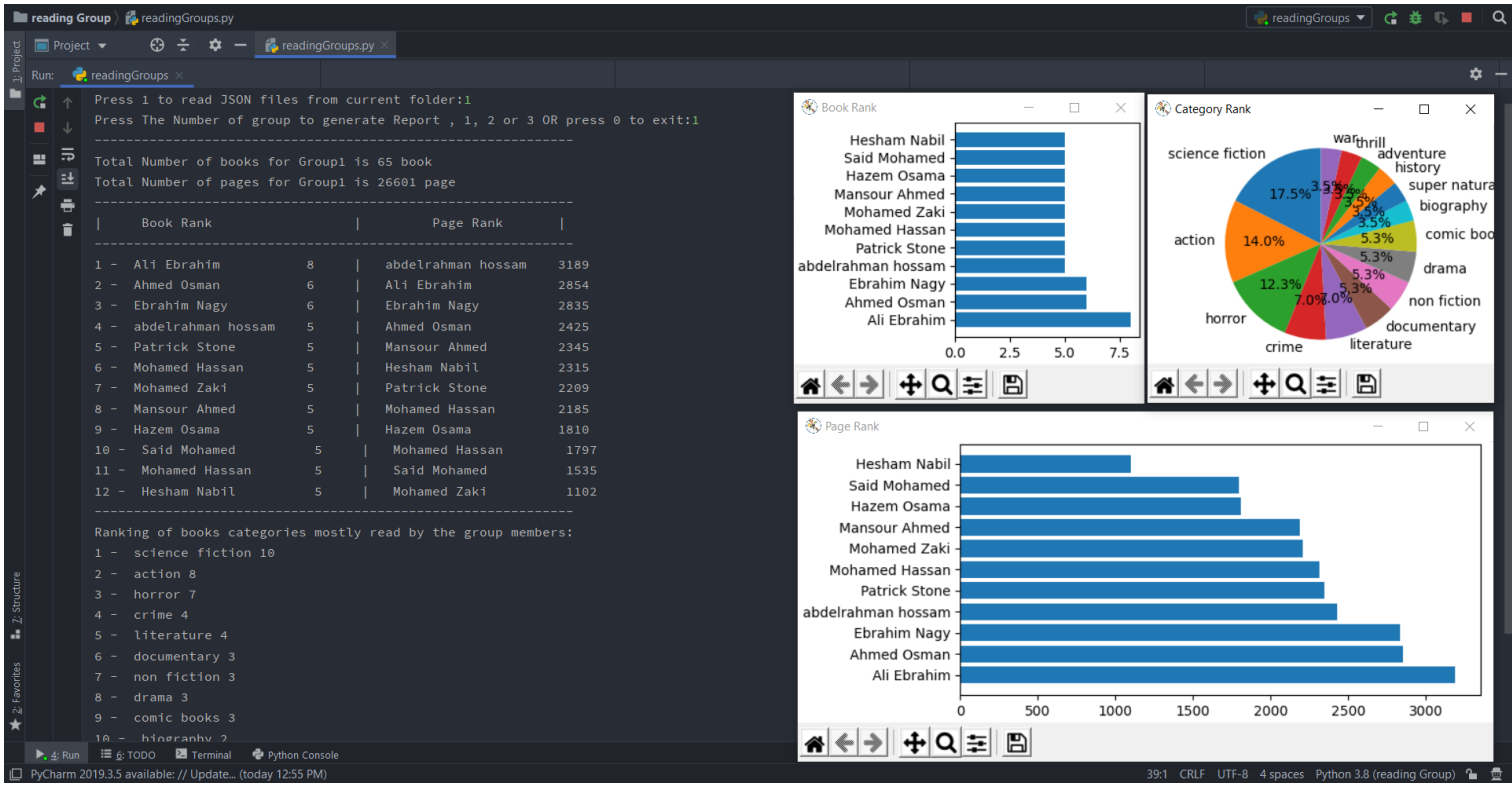
```
        return "Total Number of pages for {} is {}
page".format(self.name, total_pages)

    def get_book_categories(self):

        categories = {}
        for reader in self.readers:
            category = reader.get_category()

            for i in category:
                if i not in categories:
                    categories[i] = 1
                elif i in categories:
                    categories[i] = categories[i] + 1
        categories = sorted(categories.items(),
key=operator.itemgetter(1), reverse=True)

        return categories

    def get_user_book_rank(self):
        book_rank = []
        for reader in self.readers:
            name = reader.get_name()
            books = reader.get_no_books()
            user = (name, books)
            book_rank.append(user)
        book_ranked = sorted(book_rank, key=lambda x: x[1],
reverse=True)
        return book_ranked

    def get_user_page_rank(self):
        """
        :return: the page rank of the user
        """
        page_rank = []
        for reader in self.readers:
            name = reader.get_name()
            pages = reader.get_no_pages()
            user = (name, pages)
            page_rank.append(user)
        page_ranked = sorted(page_rank, key=lambda x: x[1],
reverse=True)
        return page_ranked


group1 = ReadingGroup('Group1', 4)
group2 = ReadingGroup('Group2', 4)
group3 = ReadingGroup('Group3', 4)


def read_files():
```

```
    """
    opens the JSON file and deals with data in it
    :return:
    """
    for filename in glob('*.json'):
        with open(filename) as f:
            data = json.load(f)  # open the json file
            # do something with the data
            user_name = data['name']
            user_number = data['mobileNumber']
            user_email = data['email']
            category = []
            no_books = 0
            no_pages = 0
            # print("after opining file: ", user_name,
user_number, user_email)

            for i in data['books']:
                no_books = no_books + 1

                for key in i:
                    if key == 'pages':
                        no_pages = no_pages + int(i[key])
                    elif key == 'category':
                        category.append(i[key])

        filename = Reader(user_name, user_number, user_email,
no_books, no_pages, category)
        if group1.add_reader(filename):
            pass
        elif group2.add_reader(filename):
            pass
        elif group3.add_reader(filename):
            pass


def report(group_no):
    """

    :param group_no: takes group_no as an instance
    :return: report with all data and ranks of users and books
    """

print("-------------------------------------------------------
--")
    print(group_no.get_total_books())
    print(group_no.get_total_pages())
    x1 = []
    y1 = []
    x2 = []
    y2 = []
    x3 = []
```

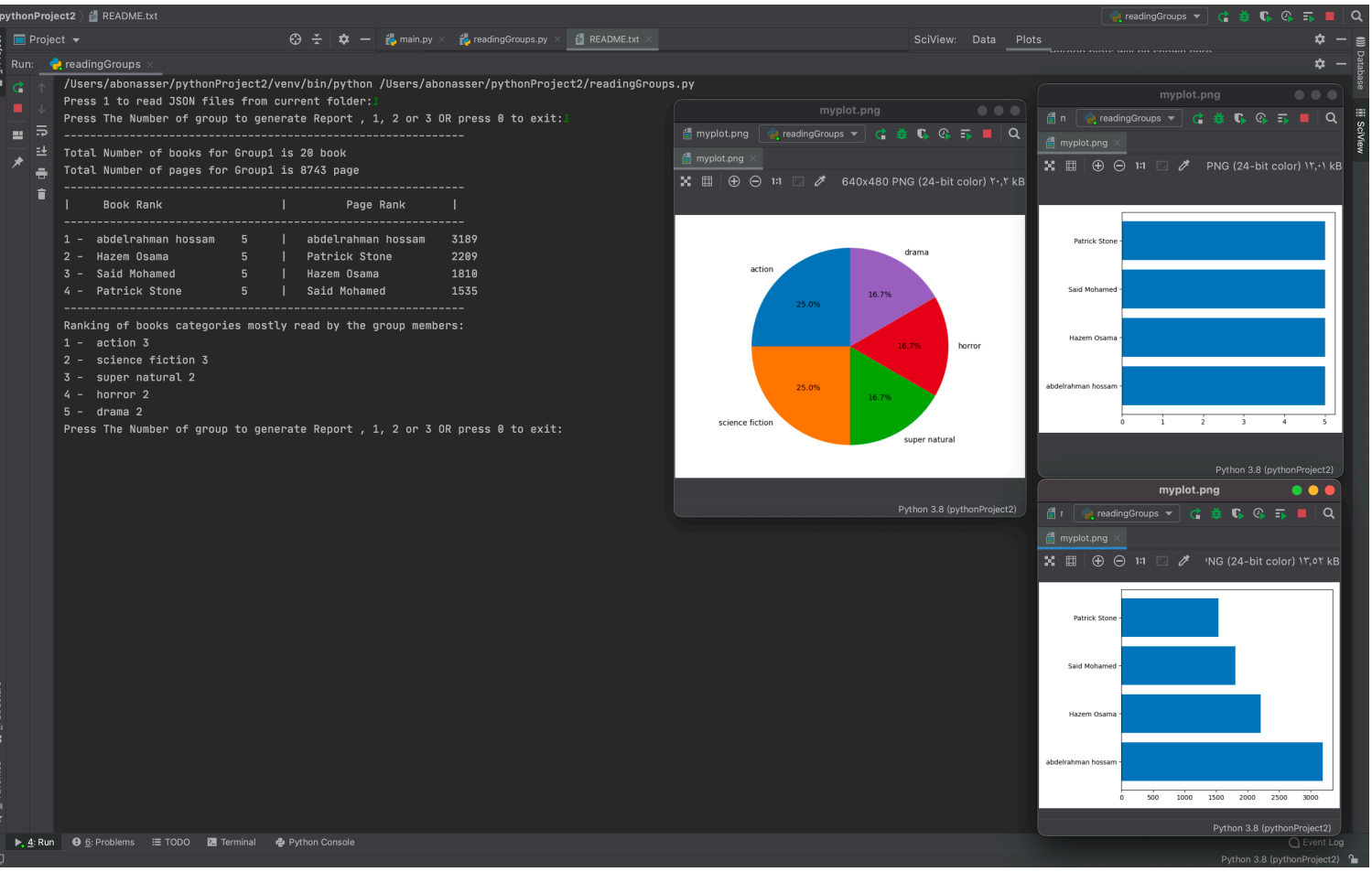
```
    y3 = []
    en = 1

print("-------------------------------------------------------------
--")
    print('|{:^20}              |    {:^20} |'.format("Book Rank",
"Page Rank"))

print("-------------------------------------------------------------
--")
    for (i, j) in zip(group_no.get_user_book_rank(),
group_no.get_user_page_rank()):
        b_rank = [i][0]
        p_rank = [j][0]

        line_new = '{:<20}  {:<4}  |   {:<20}
{:<5}'.format(b_rank[0], b_rank[1], p_rank[0], p_rank[1])
        print(en, "- ", line_new)
        en += 1
        x2.append(b_rank[0])
        y2.append(b_rank[1])
        x3.append(p_rank[0])
        y3.append(p_rank[1])

print("-------------------------------------------------------------
--")
    print("Ranking of books categories mostly read by the group
members: ")
    enu = 1

    for i in group_no.get_book_categories():
        cat = [i][0]
        if cat[1] < 2:
            continue
        else:
            print(enu, "- ", cat[0], cat[1])
            x1.append(cat[0])
            y1.append(cat[1])
        enu += 1
    # plt.plot(y1, x1, label="line 1")
    plot1 = plt.figure("Category Rank")
    plt.pie(y1, labels=x1, startangle=90, autopct='%1.1f%%')

    plot2 = plt.figure("Book Rank")
    plt.barh(x2, y2)

    plot3 = plt.figure("Page Rank")
    plt.barh(x2, y3)

    plt.show()

#_________Main()_________
```

```
group_no = ""
group_input = 1
start_input = input("Press 1 to read JSON files from current
folder:")
while group_input >= 1:
    if start_input == "1":
        read_files()
        group_input = int(input("Press The Number of group to
generate Report , 1, 2 or 3 OR press 0 to exit:"))
        if group_input == 0:
            break
        elif group_input == 1:
            group_no = eval("group" + str(group_input))
            report(group_no)
        elif group_input == 2:
            group_no = eval("group" + str(group_input))
            report(group_no)
        elif group_input == 3:
            group_no = eval("group" + str(group_input))
            report(group_no)

        else:
            print("Please enter the right group number.")
    else:break
```

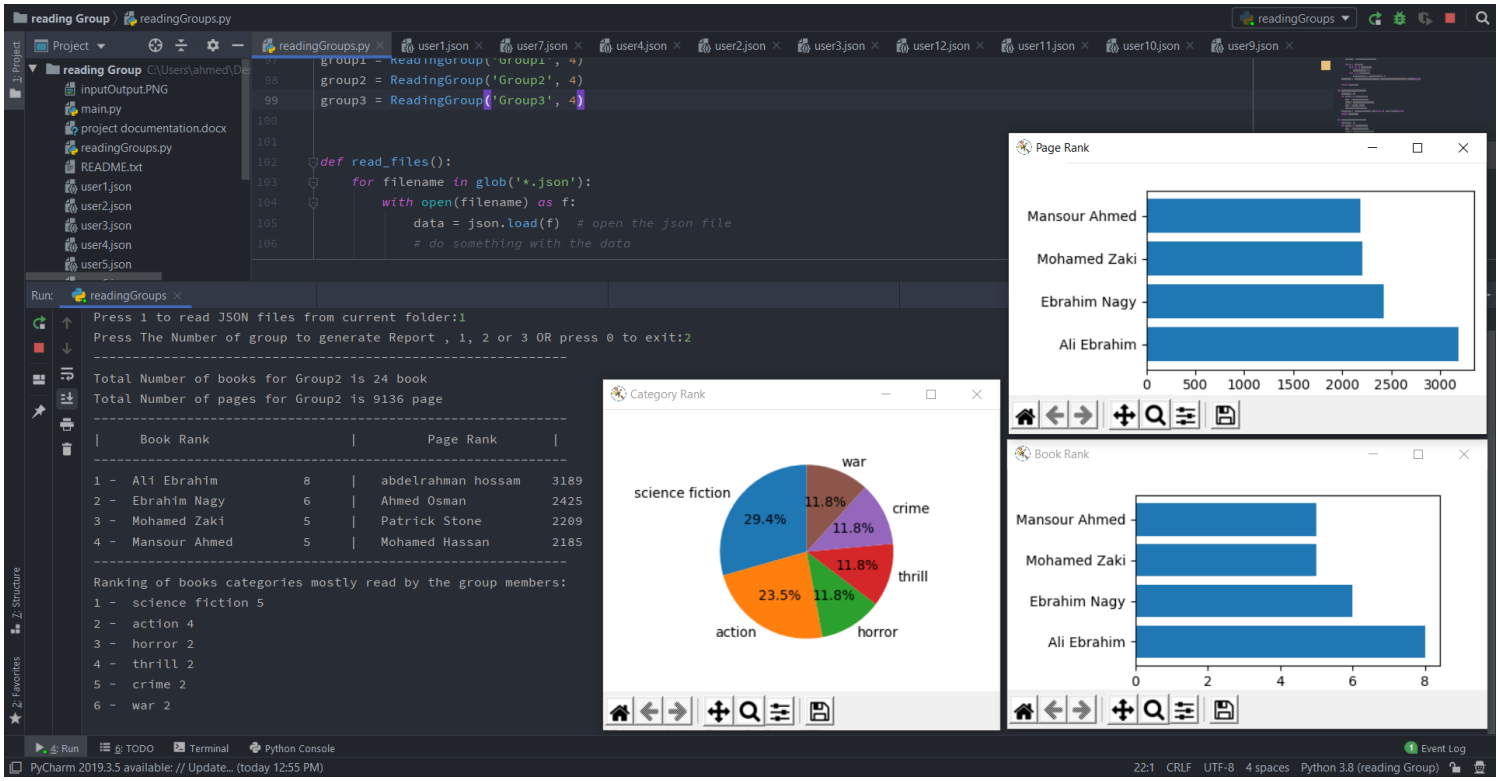
Input & Output



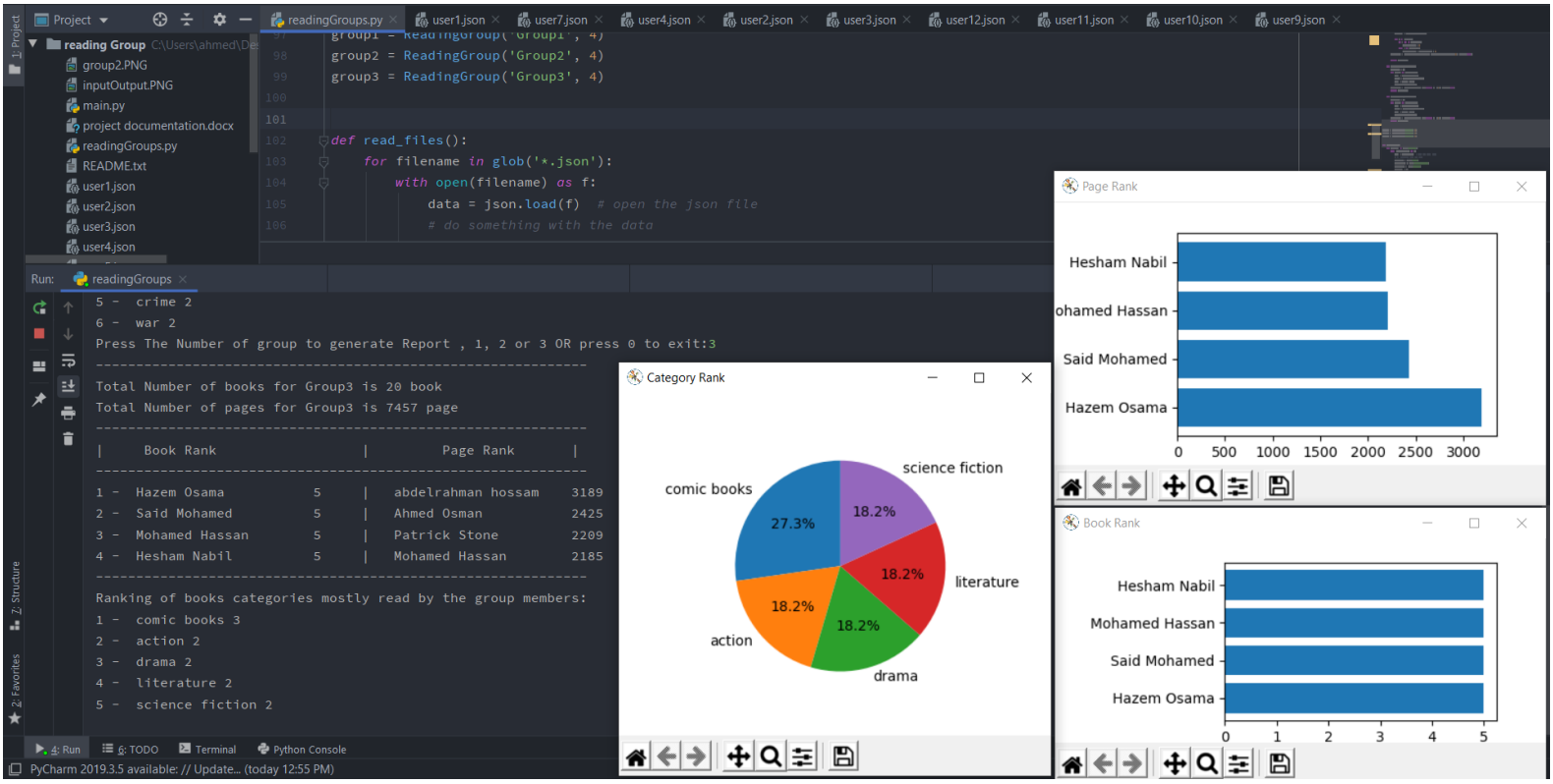
Group 1


/Users/abonasser/pythonProject2/venv/bin/python /Users/abonasser/pythonProject2/readingGroups.py
Press 1 to read JSON files from current folder:1
Press The Number of group to generate Report , 1, 2 or 3 OR press 0 to exit:1
Total Number of books for Group1 is 20 book
Total Number of pages for Group1 is 8743 page
| Book Rank | Page Rank |
1 - abdelrahman hossam 5 | abdelrahman hossam 3189
2 - Hazem Osama 5 | Patrick Stone 2209
3 - Said Mohamed 5 | Hazem Osama 1810
4 - Patrick Stone 5 | Said Mohamed 1535
Ranking of books categories mostly read by the group members:
1 - action 3
2 - science fiction 3
3 - super natural 2
4 - horror 2
5 - drama 2
Press The Number of group to generate Report , 1, 2 or 3 OR press 0 to exit:
myplot.png
640x480 PNG (24-bit color)
action
drama
horror
science fiction
super natural
25.0%
16.7%
16.7%
25.0%
16.7%
Python 3.8 (pythonProject2)
Patrick Stone
Said Mohamed
Hazem Osama
abdelrahman hossam
0
1
2
3
4
5
0
500
1000
1500
2000
2500
3000

Group 2


reading Group
readingGroups.py
group2 = ReadingGroup('Group2', 4)
group3 = ReadingGroup('Group3', 4)
def read_files():
for filename in glob('*.json'):
with open(filename) as f:
data = json.load(f) # open the json file
# do something with the data
Press 1 to read JSON files from current folder:1
Press The Number of group to generate Report , 1, 2 or 3 OR press 0 to exit:2
Total Number of books for Group2 is 24 book
Total Number of pages for Group2 is 9136 page
| Book Rank | Page Rank |
1 - Ali Ebrahim 8 | abdelrahman hossam 3189
2 - Ebrahim Nagy 6 | Ahmed Osman 2425
3 - Mohamed Zaki 5 | Patrick Stone 2209
4 - Mansour Ahmed 5 | Mohamed Hassan 2185
Ranking of books categories mostly read by the group members:
1 - science fiction 5
2 - action 4
3 - horror 2
4 - thrill 2
5 - crime 2
6 - war 2
Page Rank
Mansour Ahmed
Mohamed Zaki
Ebrahim Nagy
Ali Ebrahim
0 500 1000 1500 2000 2500 3000
Category Rank
war
science fiction
crime
thrill
action
horror
29.4%
11.8%
11.8%
11.8%
23.5%
11.8%
Book Rank
Mansour Ahmed
Mohamed Zaki
Ebrahim Nagy
Ali Ebrahim
0 2 4 6 8

Group 3


Project
reading Group C:\Users\ahmed\De
group2.PNG
inputOutput.PNG
main.py
project documentation.docx
readingGroups.py
README.txt
user1.json
user2.json
user3.json
user4.json
readingGroups.py
user1.json
user7.json
user4.json
user2.json
user3.json
user12.json
user11.json
user10.json
user9.json
group2 = ReadingGroup('Group2', 4)
group3 = ReadingGroup('Group3', 4)
def read_files():
for filename in glob('*.json'):
with open(filename) as f:
data = json.load(f) # open the json file
# do something with the data
Run:
readingGroups
5 - crime 2
6 - war 2
Press The Number of group to generate Report , 1, 2 or 3 OR press 0 to exit:3
Total Number of books for Group3 is 20 book
Total Number of pages for Group3 is 7457 page
| Book Rank | Page Rank |
1 - Hazem Osama 5 | abdelrahman hossam 3189
2 - Said Mohamed 5 | Ahmed Osman 2425
3 - Mohamed Hassan 5 | Patrick Stone 2209
4 - Hesham Nabil 5 | Mohamed Hassan 2185
Ranking of books categories mostly read by the group members:
1 - comic books 3
2 - action 2
3 - drama 2
4 - literature 2
5 - science fiction 2
4: Run
6: TODO
Terminal
Python Console
PyCharm 2019.3.5 available: // Update... (today 12:55 PM)
Category Rank
science fiction
comic books
literature
action
drama
18.2%
27.3%
18.2%
18.2%
18.2%
Page Rank
Hesham Nabil
ohamed Hassan
Said Mohamed
Hazem Osama
0
500
1000
1500
2000
2500
3000
Book Rank
Hesham Nabil
Mohamed Hassan
Said Mohamed
Hazem Osama
0
1
2
3
4
5

- **Description of all the key language features**

Class class name : **def Constructor function name:** **Class input variables** **def class method name:** **code to do something**	**Line 7 to 35**
If (statment): **Elif(statement):** **Else():**	**Used multiple times**
For(initialize limit object): **code**	**Used multiple times**
def function name: **code to do something**	**102 and 133**
Variables without declaration **total_books = 0**	**Used multiple times**
List name=[] **Multidimintional array=[[], [], []]** **page_rank = []**	**Line 39, 77, 87, 110, 137**

Dictionary={} **categories = {}**	**Line 63**
While condition : **code**	**Line 187**
Class object **group1 = ReadingGroup('Group1', 20)**	**Line 97, 98, 99**